



UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES

CARRERA DE TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

PREVIO A LA OBTENCIÓN DEL TÍTULO DE:

INGENIERA EN TELECOMUNICACIONES

**“IMPLEMENTACIÓN DE SISTEMAS DEFINIDOS POR RADIO BASADO EN
CÓDIGO ABIERTO PARA MODULACIONES PSK Y QAM EN FASE DE
CUADRATURA .”**

AUTOR:

SAENZ VALERO RUBY MONIK

DOCENTE TUTOR:

ING. LUIS MIGUEL AMAYA FARIÑO, MSC.

LA LIBERTAD, ECUADOR

2024

DECLARACIÓN DE DOCENTE TUTOR

En mi calidad de Docente Tutor del Trabajo de Integración Curricular, **“Implementación De Sistemas Definidos Por Radio Basado En Código Abierto Para Modulaciones PSK Y QAM En Fase De Cuadratura”**, elaborado por **Saenz Valero Ruby Monik**, estudiante de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Península de Santa Elena, previo a la obtención del título de Ingeniera en Telecomunicaciones, me permito afirmar que, tras supervisar el desarrollo y estructura final de su trabajo, este cumple con los estándares académicos requeridos, En consecuencia, lo considero apto en todos sus aspectos y listo para ser evaluado por el docente especialista.

Atentamente

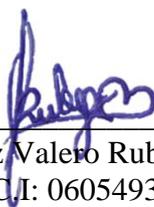


Ing. Luis Amaya Fariño, Mgtr
DOCENTE TUTOR

DECLARACIÓN AUTORIA DEL ESTUDIANTE

El presente trabajo de Integración Curricular, con el título **“Implementación De Sistemas Definidos Por Radio Basado En Código Abierto Para Modulaciones PSK Y QAM En Fase De Cuadratura”**, declaro que la concepción, análisis y resultados son originales a la actividad educativa en el área de Telecomunicaciones.

Atentamente



Saenz Valero Ruby Monik
C.I: 0605493923

DECLARACIÓN DE DOCENTE ESPECIALISTA

En mi calidad de Docente Especialista del Trabajo de Integración Curricular, **“Implementación De Sistemas Definidos Por Radio Basado En Código Abierto Para Modulaciones PSK Y QAM En Fase De Cuadratura”**, elaborado por **Saenz Valero Ruby Monik**, estudiante de la Carrera de Telecomunicaciones, Facultad de Sistemas y Telecomunicaciones de la Universidad Península de Santa Elena, previo a la obtención del título de Ingeniera en Telecomunicaciones, me permito afirmar que, tras supervisar el desarrollo y estructura final de su trabajo, este cumple con los estándares académicos requeridos. En consecuencia, lo considero apto en todos sus aspectos y listo para la sustentación del trabajo.

Atentamente



Firmado
digitalmente por
DANIEL
ARMANDO
JARAMILLO
CHAMBA

Ing. Daniel Jaramillo Chamba, M.Sc.
DOCENTE ESPECIALISTA

TRIBUNAL DE SUSTENTACIÓN



Ing. Ronald Rovira Jurado, Ph.D.
DIRECTOR DE LA CARRERA



Firmado
digitalmente por
DANIEL
ARMANDO
JARAMILLO
CHAMBA

Ing. Daniel Jaramillo Chamba, M.Sc.
DOCENTE ESPECIALISTA



Firmado
digitalmente
por LUIS
MIGUEL
AMAYA
FARINO

Ing. Luis Amaya Fariño, Mgtr.
DOCENTE TUTOR GUÍA



Ing. Corina Gonzabay De la A, Mgtr.
SECRETARÍA

DECLARACIÓN DE RESPONSABILIDAD

Quien suscribe, **Saenz Valero Ruby Monik** con **C.I:0605493923**, estudiante de la carrera de Telecomunicaciones que presento el trabajo de titulación denominado **“Implementación De Sistemas Definidos Por Radio Basado En Código Abierto Para Modulaciones PSK Y QAM En Fase De Cuadratura”** pertenece y es exclusiva responsabilidad de la autora y pertenece al patrimonio intelectual de la Universidad Estatal Península de Santa Elena.

Atentamente



Saenz Valero Ruby Monik
C.I: 0605493923

DEDICATORIA

Este logro se erige en honor a mis queridos padres, Jesús Sáenz y Mónica Valero, cuyo inquebrantable apoyo y amor han sido pilares fundamentales a lo largo de mi camino. Vuestra constante inspiración y orgullo han sido mi faro en este viaje.

A mis adoradas hermanas, Ingrid y Stefanía, así como a mi primo Joseph, quienes siempre han sido mi sostén, escuchándome y aconsejándome en cada encrucijada, guiándome siempre hacia el sendero del bien.

Con gratitud y cariño infinitos,

Ruby Monik Sáenz Valero.

AGRADECIMIENTO

En primer lugar, agradezco a Dios por el regalo de la vida y por ser mi guía constante en este camino.

A mis padres y familia, les agradezco profundamente por su inquebrantable apoyo y esfuerzo.

Le agradezco por su invaluable orientación al tutor y mentor Luis Amaya, que me acompañó en todo este proceso, al tutor especialista Daniel Jaramillo, mi gratitud por su paciencia y valiosos consejos para este proyecto.

A mis amigos Manuel, Segundo y Cristián, les agradezco por haber compartido sus conocimientos y experiencias.

A mi querido amigo Benjamín B., le agradezco por hacer esta carrera tan amena y agradable.

Y finalmente, a mi mejor amigo Jimmy Guillin C., gracias por los momentos increíbles compartidos y por ser un constante impulso para seguir adelante.

Vuestra presencia ha sido fundamental en este viaje.

Ruby Monik Sáenz Valero.

Resumen

Hoy en día las transmisiones digitales son las más utilizadas debido a que en cuanto a costos, velocidad, consumo de energía, ancho de banda y optimización, presentan mayores ventajas que las transmisiones analógicas porque son más resistentes al ruido y pueden manejar múltiples señales digitales multicanal.

El análisis del comportamiento de las modulaciones PSK y QAM dentro de un medio inalámbrico al momento de presentar algún tipo de pérdida de información cuando inicia una transmisión o una recepción por cualquier motivo detallado al inicio de la presentación en procesos de estudio dentro del ámbito educativo, donde en la actualidad los niveles de enseñanza en ciertos casos vienen limitados por una razón u otra.

El desarrollo del presente proyecto se enfoca en la presentación de nuevas medidas de estudio en este caso dando a conocer el software GNU RADIO que por medio del dispositivo SDR (HackRF One) incentivando el uso de este con fines educativos dentro de la malla curricular en las asignaturas de comunicaciones digitales, procesamiento digital de señales correspondiente a la carrera de telecomunicaciones dentro de la Universidad Estatal Península de Santa Elena.

Palabras Clave: Psk, Qam, Gnu Radio, HackRF One, Sdr.

Abstract

Today digital transmissions are the most used because in terms of costs, speed, energy consumption, bandwidth, and optimization, they have greater advantages than analog transmissions, since they have greater immunity to noise, they can be processed and multi-channel greater number of digital signals.

The analysis of the behavior of the PSK and QAM modulations within a wireless medium at the moment of presenting some type of loss of information when a transmission or reception begins for any reason detailed at the beginning of the presentation in study processes within the educational field, where currently the levels of education in certain cases are limited for one reason or another.

The development of this project focuses on the presentation of new study measures in this case making known the GNU radio software that through the SDR device (HackRF one) encouraging the use of this for educational purposes within the curricular mesh in the subjects of digital communications, digital signal processing corresponding to the telecommunications career within the Santa Elena Peninsula State University.

Keywords: Psk, Qam, Gnu Radio, HackRF One, Sdr

INDICE

Tabla de contenido

DECLARACIÓN DE DOCENTE TUTOR	II
DECLARACIÓN AUTORIA DEL ESTUDIANTE	III
DECLARACIÓN DE DOCENTE ESPECIALISTA	IV
TRIBUNAL DE GRADO	V
DECLARACIÓN DE RESPONSABILIDAD	VI
DEDICATORIA	VII
AGRADECIMIENTO	VIII
RESUMEN	IX
ABSTRACT	X
ÍNDICE DE ABREVIATURAS	XIV
ÍNDICE DE IMÁGENES	XVI
ÍNDICE DE TABLA	XX
ÍNDICE DE ECUACIONES	XX
ÍNDICE DE ANEXO	XX
IMPLEMENTACIÓN DE SISTEMAS DEFINIDOS POR RADIO BASADO EN CÓDIGO ABIERTO PARA MODULACIONES PSK Y QAM EN FASE DE CUADRATURA	XXI
INTRODUCCIÓN	22
CAPÍTULO I : EL PROBLEMA DE LA INVESTIGACIÓN	24
1.1 PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA	24
1.2 OBJETIVOS	25
1.2.1 <i>Objetivo General</i>	25

<i>1.2.2 Objetivo Especifico</i>	26
1.3 JUSTIFICACIÓN	26
1.4 ANTECEDENTES DE INVESTIGACIÓN	27
CAPÍTULO 2: MARCO CONCEPTUAL	29
2.1 FUNDAMENTO TEÓRICO O CONCEPTUAL SOBRE MODULACIÓN	29
2.2 MODULACIÓN DIGITAL	31
2.3 MODULACIÓN PSK (PHASE SHIFT KEYING)	33
<i>2.3.1 Modulación Por Desplazamiento De Fase En Binaria BPSK (2-PSK)</i>	34
<i>2.3.2 Modulación Por Desplazamiento De Fase En Cuadratura QPSK (4-PSK)</i>	34
<i>2.3.3 Modulación Por Desplazamiento De Fase En (8-PSK)</i>	35
<i>2.3.4 Modulación Por Desplazamiento De Fase En (16-PSK)</i>	36
2.4 MODULACIÓN QAM (QUADRATURE AMPLITUDE MODULATION)	38
<i>2.4.1 Aplicación De La Modulación QAM</i>	39
<i>2.4.2 Modulación De Amplitud En Cuadratura En (8-QAM)</i>	40
<i>2.4.3 Modulación De Amplitud En Cuadratura En (16, 64, 256 -QAM)</i>	40
2.5 SOFTWARE DEFINIDO POR RADIO (SDR)	42
<i>2.5.1 Esquema Genérico Del SDR</i>	46
<i>2.5.2 Características Del Hackrf One</i>	49
<i>2.5.3 El Concepto de Arquitectura de SDR</i>	50
<i>2.5.4 Arquitectura de Rf para SDR</i>	51
2.6 UNIVERSAL SOFTWARE RADIO PERIPHERICAL (USRP)	53
<i>2.6.1 Comparación de SDR Vs Hackrf One</i>	54
CAPÍTULO 3: METODOLOGÍA	56
3.1 HARDWARE PARA SDR	59
3.1.1 MEDIOS COMPUTACIONALES NECESARIOS EN EL PROCESO SDR	60
3.2 SOFTWARE PARA SDR	62
3.2.1 GNU RADIO	62

3.3 INSTALACIÓN GNU RADIO	64
3.4 TIPOS DE DATOS EN GNU RADIO.....	73
3.5 MATERIALES.....	73
3.6 DESARROLLO DE LA PROPUESTA.....	74
3.6.1 TRANSMISIÓN DE PSK.....	74
3.6.2 TRANSMISIÓN DE QAM.....	89
3.6.3 MODULACIÓN DE 32-QAM.....	94
3.6.4 RECEPCIÓN	99
3.6.5 MODULACIONES PSK EN SIMULINK	105
3.6.6 MODULACIONES QAM EN GUIDE.....	110
CAPÍTULO 4: ANÁLISIS DE RESULTADOS, CONCLUSIÓN Y RECOMENDACIÓN.	121
4.1 ANÁLISIS DE LOS RESULTADOS.....	121
4.2 CONCLUSIÓN.....	123
4.3 RECOMENDACIÓN	124
MATERIALES DE REFERENCIAS.....	124
BIBLIOGRAFÍA.....	124
ANEXOS	131

Índice De Abreviaturas

ADC: Convertidores De Analógico A Digital.

AM: Modulación de amplitud.

ASK: Codificación por desplazamiento de amplitud.

BER: Tasa de error de bit.

BPSK: Modulación Por Desplazamiento De Fase Binario.

DAC: Convertidores De Digital A Analógico.

DSP: Procesamiento De Señal Digital.

FM: Modulación de frecuencia.

FSK: Codificación por desplazamiento de frecuencia.

LSB: Byte menos significativo.

MIMO: Multiple Input Multiple Output.

MSB: Byte más significativo.

PM: Modulación de fase.

PSK: Phase Shift Keying (Modulación Por Desplazamiento De Fase).

QAM: Quadrature Amplitude Modulation (Modulación De Amplitud En Cuadratura).

QoS: Calidad de servicio.

QPSK: Quadrature Phase-Shift Keying (Modulación Por Desplazamiento De Fase Cuaternaria).

Rad: Radián.

RF: Radiofrecuencia.

Rx: Recepción.

SDR: Software Defined Radio (Software Definido Por Radio).

SNR: Relación Señal/Ruido.

Tx: Transmisión.

ZIF: Zero-IF Architecture.

Índice De Imágenes

<i>Figura. 1 Sistema de comunicación digital</i>	30
<i>Figura. 2 Método de Modulación</i>	30
<i>Figura. 3 Modulaciones digitales</i>	31
<i>Figura. 4 Modulación PSK</i>	33
<i>Figura. 5 Modulación BPSK</i>	34
<i>Figura. 6 Constelación QPSK</i>	35
<i>Figura. 7 Constelación 8-PSK</i>	36
<i>Figura. 8 Diagrama de constelación 16-PSK</i>	37
<i>Figura. 9 Constelaciones QAM rectangulares</i>	39
<i>Figura. 10 Modulación De 8-QAM</i>	40
<i>Figura. 11 Constelaciones de 16, 32, 64-QAM</i>	41
<i>Figura. 12 Sistema De Comunicación Ideal</i>	44
<i>Figura. 13 Diagrama genérico de los bloques electrónicos que componen un SDR</i>	46
<i>Figura. 14 Placa Impresa del HackRF One</i>	48
<i>Figura. 15 Antena ANT500</i>	49
<i>Figura. 16 lóbulos de radiación</i>	50
<i>Figura. 17 Equipo USRP</i>	53
<i>Figura. 18 Conexión del módulo</i>	56
<i>Figura. 19 Diagrama de la propuesta</i>	58
<i>Figura. 20 Equipo HackRF One</i>	59
<i>Figura. 21 Venta de trabajo en Gnu Radio</i>	62
<i>Figura. 22 Arquitectura Gnu Radio</i>	63
<i>Figura. 23 Carpeta de instaladores en OneDrive</i>	64
<i>Figura. 24 vc_redist.x86.exe</i>	65
<i>Figura. 25 Python-2.7.15.amd64.msi</i>	65
<i>Figura. 26 Pantalla del Sistema</i>	66
<i>Figura. 27 Configuración avanzada del sistema y Variable de entorno</i>	66
<i>Figura. 28 Variables del sistema</i>	67
<i>Figura. 29 Pantalla de símbolo del sistema (CMD)</i>	68

<i>Figura. 30</i> CMD Python.....	68
<i>Figura. 31</i> Python get-pip.py.....	69
<i>Figura. 32</i> Archivos (.whl).....	70
<i>Figura. 33</i> uhd_3.9.6-release_x64_VS2015.exe.....	70
<i>Figura. 34</i> Instalador de Gnu Radio.....	71
<i>Figura. 35</i> Aplicación Gnu Radio Companion.....	71
<i>Figura. 36</i> Configuración del entorno de Gnu Radio.....	72
<i>Figura. 37</i> Entorno gráfico de Gnu Radio.....	72
<i>Figura. 38</i> diagrama de transmisor para modulaciones de 8-PSK.....	74
<i>Figura. 39</i> Variables y librerías de PSK.....	75
<i>Figura. 40</i> Bloque Options y Propiedades.....	75
<i>Figura. 41</i> Bloque Variable y Propiedades.....	76
<i>Figura. 42</i> Bloque Wx Gui Slider y Propiedades.....	76
<i>Figura. 43</i> FI_Gain y Propiedades.....	77
<i>Figura. 44</i> Ganancia de la Banda Base y Propiedades.....	77
<i>Figura. 45</i> Ganancia de Radio frecuencia (RF) y Propiedades.....	78
<i>Figura. 46</i> Bloque Wx Gui Text Box y Propiedades.....	78
<i>Figura. 47</i> Bloque Wav File Source y Propiedades.....	79
<i>Figura. 48</i> Bloque Multiply Const y Propiedades.....	79
<i>Figura. 49</i> Bloque Rational Resampler y Propiedades.....	79
<i>Figura. 50</i> Bloque WBFM Transmit y Propiedades.....	80
<i>Figura. 51</i> Bloque Low Pass Filter y Propiedades.....	80
<i>Figura. 52</i> Bloque Multiply Const con complex y Propiedades.....	81
<i>Figura. 53</i> Bloque Osmocom Sink y Propiedades.....	81
<i>Figura. 54</i> Bloque Float To Short y Propiedades.....	82
<i>Figura. 55</i> Bloque CODEC2 Audio Encoder y Propiedades.....	82
<i>Figura. 56</i> Bloque Vector to Stream y Propiedades.....	82
<i>Figura. 57</i> Bloque Packet Encoder y Propiedades.....	83
<i>Figura. 58</i> Bloque PSK Mod y Propiedades.....	83
<i>Figura. 59</i> Bloque Throttle y Propiedades.....	84

<i>Figura. 60</i> Bloque WX GUI FFT Sink y Propiedades.....	84
<i>Figura. 61</i> Resultado de Constelaciones a) 4-PSK. b) 8- PSK. c) 16- PSK.....	85
<i>Figura. 62</i> Diagrama de transmisión de las modulaciones 16 QAM.	89
<i>Figura. 63</i> Variable Símbolo_QAM y Propiedades.	89
<i>Figura. 64</i> Variable Bits/Símbolo y Propiedades.	90
<i>Figura. 65</i> Variable Muestras/Símbolo y Propiedades.	90
<i>Figura. 66</i> Bloque WX GUI Notebook y Propiedades.	90
<i>Figura. 67</i> Bloque WX GUI Scope Sink y Propiedades.	91
<i>Figura. 68</i> Diagrama de Constelaciones resultado a) 16-QAM. b) 64-QAM.....	92
<i>Figura. 69</i> Diagrama de la modulación de 32-QAM.....	94
<i>Figura. 70</i> Bloque Random Source y Propiedades.	94
<i>Figura. 71</i> Bloque Packed to Unpacked y Propiedades.....	95
<i>Figura. 72</i> Bloque Chunks to Symbols y Propiedades.	95
<i>Figura. 73</i> Bloque QT GUI Constellation Sink y Propiedades.	96
<i>Figura. 74</i> Diagrama De Constelación con posición en I y Q.	98
<i>Figura. 75</i> Diagrama De Constelación, Resultado De 32-QAM Ideal.....	99
<i>Figura. 76</i> Receptor.....	100
<i>Figura. 77</i> Bloque Osmocom Sourse y Propiedades.....	101
<i>Figura. 78</i> Bloque Signal Source y Propiedades.	101
<i>Figura. 79</i> Bloque Multiply y Propiedades.	102
<i>Figura. 80</i> Bloque WBFM Receiver y Propiedades.	102
<i>Figura. 81</i> Bloque Audio Sink y Propiedades.....	103
<i>Figura. 82</i> Resultado de recepción en la frecuencia de 104.3Mhz.....	103
<i>Figura. 83</i> Modulaciones 4,8,16-PSK en Simulink.	105
<i>Figura. 84</i> Bloque Random Integer y Parámetros.	106
<i>Figura. 85</i> Bloque QPSK Modulador y Parámetros.....	106
<i>Figura. 86</i> Bloque 8-PSK Modulador y Parámetros.	107
<i>Figura. 87</i> Bloque AWGN y Parámetros.....	107
<i>Figura. 88</i> Bloque QPSK Demodulador y Parámetros.....	108
<i>Figura. 89</i> Bloque Error Rate y Parámetros.	108

Figura. 90 Diagrama de constelación a) 4-PSK. b) 8-PSK. c) 16-PSK con $E_b/N_0 = 10\text{dB}$	109
Figura. 91 a) Icono de MATLAB. b) Command window.....	110
Figura. 92 Ventana de guide.....	110
Figura. 93 Selector de modulaciones y codificación.	111
Figura. 94 Selector de Banda y codificación.....	113
Figura. 95 Selector de Tasa de bit y codificación.....	113
Figura. 96 Beamforming.	114
Figura. 97 a) Número de Canales. b) Número de símbolo por trama y codificación.	114
Figura. 98 Botón Iniciar y codificación.	115
Figura. 99 Resultados se las modulaciones a) 8-QAM. b) 16-QAM. c) 32-QAM. d) 64-QAM en Guide Matlab.	117
Figura. 100 Botón Salir y codificación.	120

Índice De Tabla

<i>Tabla 1</i> Comparativa de bits de modulaciones.....	32
<i>Tabla 2</i> Comparación entre RTL-SDR, HackRF One y USRP B210.....	54
<i>Tabla 3</i> Características del HackRF One.....	59
<i>Tabla 4</i> Tipos de datos en Gnu Radio.	73
<i>Tabla 5</i> Materiales.....	73
<i>Tabla 6</i> Modulación 32-QAM	96

Índice De Ecuaciones

<i>Ecuación 1</i>	32
<i>Ecuación 2</i>	38
<i>Ecuación 3</i>	38
<i>Ecuación 4</i>	87
<i>Ecuación 5</i>	87
<i>Ecuación 6</i>	96

Índice De Anexo

<i>Anexo 1: Sílabo de comunicaciones digitales 2023-2</i>	131
<i>Anexo 2: Sílabo de Procesamiento digital de señales 2023-2</i>	139
<i>Anexo 3: Factura de HackRF One</i>	145
<i>Anexo 4: Factura de computadora completa</i>	146
<i>Anexo 5 :Código de modulaciones QAM en MATLAB</i>	147
<i>Anexo 6 Guía Didáctica de Comunicación digital</i>	174
<i>Anexo 7 Guía Didáctica de Procesamiento de señales digitales</i>	184

Implementación De Sistemas Definidos Por Radio
Basado En Código Abierto Para Modulaciones Psk Y Qam
En Fase De Cuadratura

Introducción

Implementación De Sistemas Definidos Por Radio Basado En Código Abierto Para Modulaciones PSK y QAM En Fase De Cuadratura

Considerando que hoy en día la tecnología va evolucionando con nuevas técnicas aplicadas a distintas áreas como una de ellas son las telecomunicaciones dónde se enfoca en las comunicaciones inalámbricas en la cual se centra en la pérdida de información por diferentes factores como son atenuación, distorsión e interferencia donde se enfoca en la necesidad de analizar el comportamiento de las modulaciones al presentar un tipo de pérdida de información, estas consecuencias pueden ser examinadas mediante los dos tipos de software licenciados que son utilizados con mayor frecuencia como lo es Matlab es una plataforma basada en líneas de código utilizada para analizar datos profesionalmente [1], y su complemento Simulink y también se encuentra software de acceso libre como lo es GNU radio son bloques que se en cargan del procesamiento de señales, Se puede emplear con hardware de RF externo [2] y equipos físicos profesionales dedicados netamente al estudio del comportamiento de las modulaciones, los cuales dentro de un ámbito educativo son difíciles de acceder o presentan limitaciones, en el caso de software y no cuentan con una licencia que permita su uso en totalidad, debido a estas limitaciones se crean dispositivos accesibles basados en las tecnologías SDR, donde el HackRF One trabaja con frecuencia de 1 MHz a 6 GHz y es capaz de transmitir o recibir señales de radio. [3] Es un dispositivo compatible a un software libre como Gnu Radio al querer realizar un análisis en las modulaciones no obstante el dispositivo puede ser establecido para distintos usos académicos, el módulo educativo a implantar estaría enfocado para un medio de capacitación y estudio a una enseñanza híbrida,

mediante software de acceso libre y hardware que se complementen al momento de estudiar las modulaciones sin que afecten los resultados a los que se quiere llegar, tomando en cuenta que lo desarrollado en un software licenciado puede ser elaborado en un software libre sin ningún problema.

CAPÍTULO I : El Problema De La Investigación

1.1 Planteamiento y Formulación del Problema

La Real Academia de la Lengua define al término telecomunicación como el sistema de transmisión y recepción de señales, a través de un medio electromagnético [4]. Actualmente, la tecnología SDR ha evolucionado a pasos agigantados, una de ellas es la modulación digital, que es la base fundamental de transmisión de datos, donde se encuentran señales moduladas como PSK y QAM, estas se utilizan ampliamente en todo tipo de comunicaciones de datos cableadas e inalámbricas, ya que brindan mejores beneficios y es más eficiente, estas modulaciones son consideradas de gran importancia a nivel mundial, ya que posibilita el funcionamiento de equipos electrónicos como la radio, televisión, telefonía e incluso el internet [5], a su vez, siendo la base principal de la civilización que hoy en día concebimos [6].

La comunicación requiere de tres estados básicos como lo son un emisor, el mensaje y el receptor, la telecomunicación implica de todo un bagaje adicional de recursos físicos y digitales que componen y hacen aún más complejo al canal, transmitiendo el mensaje con la mayor fidelidad posible [7], sin embargo, existen elementos externos que interfieren en este proceso, debilitando, modificando, o incluso anulando el mensaje a esto se lo conoce como perturbación en la transmisión [8] .

En la actualidad la importancia de la educación en comunicaciones digitales y modulaciones es fundamental, ya que, el mundo cada vez está más conectado y es dependiente de la tecnología. El constante cambio de la tecnología requiere profesionales capacitados en Software Defined Radio (SDR), GNU Radio y dispositivos innovadores como HackRF One. Estas habilidades son necesarias no sólo para desarrollar y mantener sistemas de comunicaciones eficaces, sino también para garantizar un entorno digital.

La educación en estos campos no sólo permite a las personas mantenerse al tanto de las últimas tendencias tecnológicas, sino que también contribuye a un mayor desarrollo de la sociedad en la era de la información.

Con base en los conocimientos alcanzados en las universidades, la necesidad de adquirir nuevos elementos educativos en las asignaturas de comunicaciones digitales, procesamiento digital de señales es crucial para comprender la complejidad de los sistemas basados en las modulaciones PSK y QAM, por lo que es necesario el análisis de bloques en códigos abiertos, junto con el uso de la tecnología SDR, ya que, plantea importantes desafíos para los estudiantes que pretenden comprender estos conceptos.

Por lo tanto, es necesario abordar este problema realizando una consulta detallada y sistemática de los contenidos temáticos del sílabo en las modulaciones PSK y QAM de las asignaturas de comunicaciones digitales y procesamiento digital de señales en la carrera de telecomunicaciones en **Anexo 1** y **Anexo 2**. Además, indagar el contenido teórico asociado a cada uno de los conceptos relevantes para la ejecución exitosa y el desarrollo de la guía didáctica con práctica experimental correspondiente, con el fin de que facilite la comprensión y aplicación de estas modulaciones con el SDR y al mismo tiempo que se diseñan bloques lógicos en GNU Radio para analizar de manera práctica las modulaciones en el ámbito de las telecomunicaciones.

1.2 Objetivos

1.2.1 Objetivo General

Implementar una guía didáctica con práctica experimental que permita analizar un audio mediante un software de código abierto donde se analizará los sistemas de modulación PSK y QAM empleando tecnología SDR.

1.2.2 Objetivo Especifico

- Analizar los conceptos necesarios con información teórica relevante y proporcionar los recursos necesarios que respalde a la guía didáctica.
- Determinar las variables adecuadas para las modulaciones por desplazamiento de fase y amplitud en cuadratura, mediante el análisis de bloques lógicos aplicando sistemas SDR.
- Diseñar los bloques lógicos en el software de código abierto Gnu radio para la construcción del módulo educativo.
- Desarrollar una guía práctica basada en sistemas definidos por radio aplicados a software de código abierto para las asignaturas de comunicaciones digitales, procesamiento digital de señales.

1.3 Justificación

El análisis del comportamiento de las modulaciones PSK y QAM dentro de un medio inalámbrico, al momento de presentar algún tipo de pérdida de información cuando inicia una transmisión o una recepción, por cualquier motivo detallado al inicio de la presentación en procesos de estudio dentro del ámbito educativo, donde en la actualidad los niveles de enseñanza van enfocados a las actualizaciones de la metodología de la educación, por lo cual, se requiere ciertos niveles de conocimientos para poder analizar de mejor manera este tipo de modulaciones que se dan en las materias, las falencias de estos procesos se deban a razones económicas o en gran parte a la falta de equipamientos por parte de las instituciones educativas.

El desarrollo del presente proyecto se enfoca en la presentación de nuevas alternativas didácticas de estudio para el proceso de aprendizaje, con base en los sílabos de las asignaturas de comunicaciones digitales y procesamiento digital de señales, en la cual, el desarrollo del

aprendizaje se enfoca en analizar los procesos que se llevan a cabo sobre las señales digitales durante su transmisión a través de un sistema de comunicación digital, también se indica en el sílabo que la metodología de desarrollo en la que se va a enfocar es en Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo, la problemática de esto es que no se enfocan en una enseñanza didáctica, se justifica mediante este tipo de proyectos ya que de esa manera se dinamiza y se ilustra de mejor manera la sistematización de los procesos de modulaciones.

1.4 Antecedentes De Investigación

El desarrollo de nuevas tecnologías involucradas a la implementación de sistemas de comunicaciones inalámbricas mediante el uso de modulaciones PSK y QAM conlleva un proceso de estudio y análisis técnico por ende en la actualidad el uso de software de acceso libre tal como es GNU Radio donde interviene la tecnología SDR o softwares licenciados que permitan trabajar con este tipo señales digitales ya sea en procesos educativos o profesionales.

Desde el desarrollo de la tecnología SDR (Software Definida por Radio) daba con propósitos militares en Estados Unidos con el fin de aplicar varias tecnologías de comunicación inalámbrica en su proyecto llamado SpeakEasy [9], donde se aplicaba a bajas frecuencias de 2MHz a 200MHz con el fin de obtener un equipo programable que puede ser utilizado con estándares de comunicación futuros, de tal manera que hasta la actualidad se desarrollado a mayor magnitud este tipo de dispositivos basados en dicha tecnología que permitan trabajar a una frecuencia mayor a la de sus inicios.

Mediante un estudio realizado por Daniel Nieva y José Sanabria en el año 2018 con la temática de “Sistemas de Comunicación SDR para modulaciones PSK y QAM”

se demuestra el proceso de estudio de este tipo de modulaciones por medio de un software abierto y el uso de un hardware el cual en este caso un USRP1. [10]

El análisis en HF (High Frequency) de banda estrecha con base en el estudio del comportamiento de las modulaciones QAM y PSK realizado por Ahmaid Ismail y K. Mohamend-Pour en el año 2019 propuso la comparativa de dichas modulaciones con el fin de interpretar y determinar qué tipo de modulación es conveniente usar al momento de transmitir datos por diferentes canales dentro de un ambiente simulado. [11]

Verónica Toro 2017 en su estudio realizado por medio de un algoritmo donde establece parámetros estudiando modulaciones de una señal la cual es receptada por un SDR presento la conexión de varios dispositivos de manera simultánea aplicando métodos de modulación que hagan posible dicha acción. [12]

El diseño de un módulo que permita el uso correcto y eficaz al momento de modular y demodular QAM por medio del software GNU radio realizado por Orlando Arboleda Molina durante el año 2016 brinda una iniciativa al momento del desarrollo de diagramas dentro del software aplicando la tecnología SDR para demostrar en base de análisis los resultados obtenidos. [13]

CAPÍTULO 2: Marco Conceptual

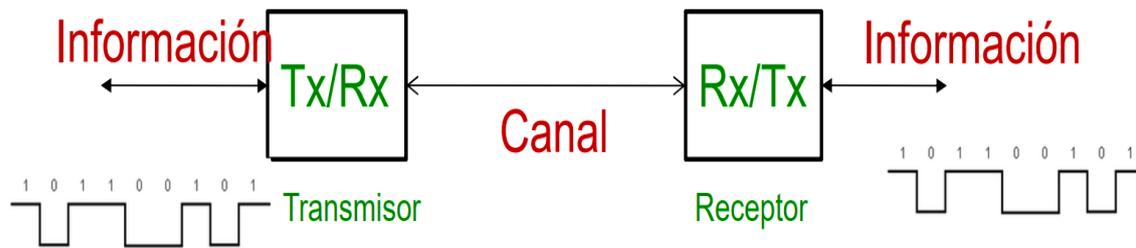
En el presente capítulo, desempeña un papel crucial al proporcionar un marco conceptual sólido y bien fundamentado para el resto del trabajo de grado. La delimitación y revisión teórica facilitará la identificación y comprensión de la información relevante para el estudio, lo que aumentará el éxito y la calidad del proyecto de investigación.

Finalmente, se establecerá una conexión clara entre las bases teóricas y el enfoque metodológico que se utilizará en la investigación. Para garantizar la coherencia y relevancia de la recopilación y el análisis de la información, es esencial establecer una conexión que se llevará a cabo en las fases posteriores del proyecto.

2.1 Fundamento Teórico o Conceptual Sobre Modulación

En el campo de las Telecomunicaciones, la modulación se refiere al conjunto de métodos utilizados para transmitir información a través de una onda portadora, que generalmente es una onda senoidal [14]. Teniendo en cuenta el concepto básico de la comunicación digital, es el traslado de información punto a punto sobre canales digitales o analógicos y puede ser representado como se observa en la **Figura. 1**, [15] donde el transmisor (Tx) es el que modifica la información original haciéndola adecuada para su transmisión, en la cual el receptor (Rx) finalmente debe convertir la información recibida a su forma original para que pueda ser enviada a su destino y a su vez procesada [16].

Figura. 1 Sistema de comunicación digital.

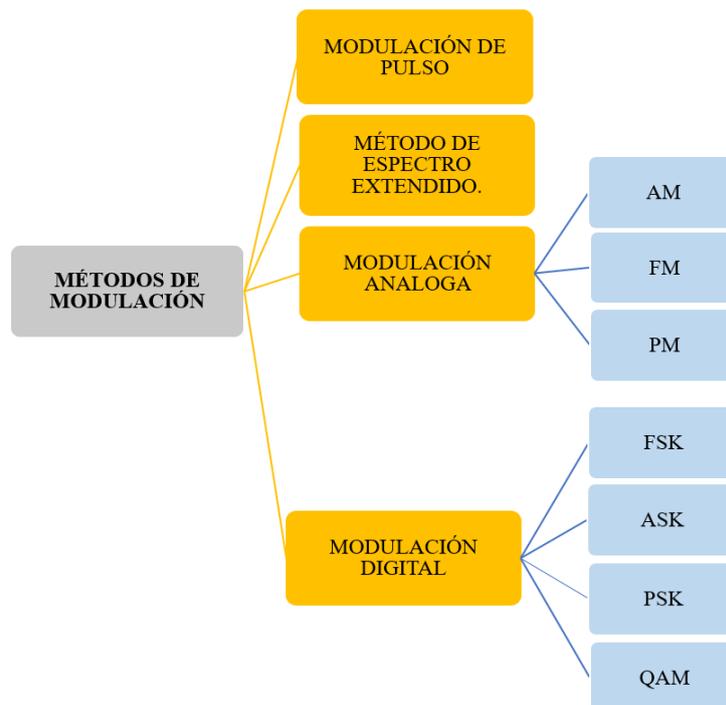


Nota. Se aprecia la transmisión y recepción de una señal digital que pasa por medio del canal transmitiendo información [17].

Estas comunicaciones mejoran el uso del canal, es decir, la modulación es el proceso de convertir datos en señales eléctricas que se pueden transmitir de manera más eficiente y mejoran el uso del canal de comunicación.

Aproximadamente, hay cuatro categorías principales de modulación, así como se observa en la **Figura. 2**:

Figura. 2 Método de Modulación



Nota. Elaborador por el autor.

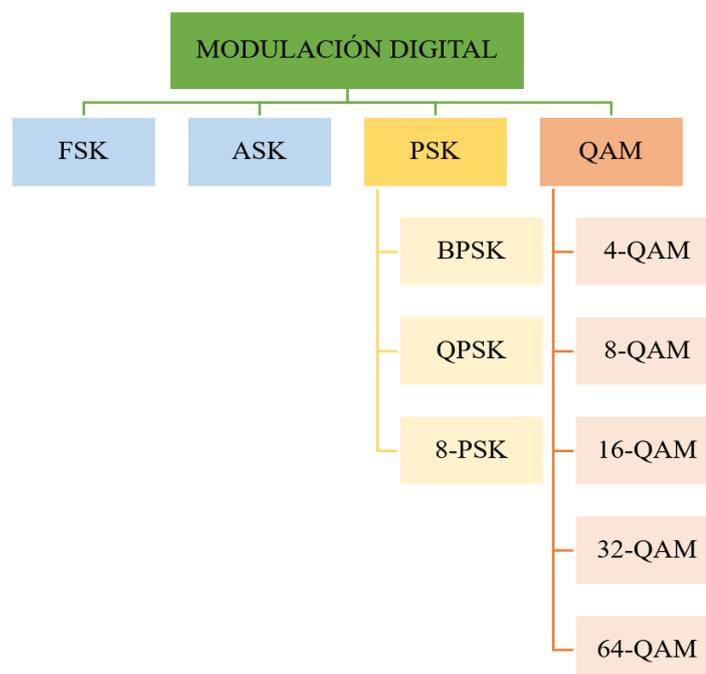
El método de espectro extendido es el que amplía la energía de la señal sobre una banda ancha, y la técnica de modulación de pulso cambian el ancho de pulso.

La modulación analógica se la encuentra en la transmisión de radio AM y FM que son de onda corta. Además, la modulación digital transmite señales binarias entre 0 y 1. Esta es una modulación de portadora única, por lo que la portadora ocupa todo el ancho de banda [18].

2.2 Modulación Digital

Las modulaciones digitales implican el procesamiento de algunos parámetros en una onda portadora en función de una señal conocida como onda moduladora, que es una señal digital binaria de "1" y "0" que transporta información, para que la señal pueda ser transmitida por un canal. Como se observa en la **Figura. 3** Modulaciones digitales., existen diferentes tipos de modulación digital [19].

Figura. 3 Modulaciones digitales.



Nota. Elaborador por el autor.

Se presenta la clasificación de la modulación para PSK y QAM. Se evaluó el desempeño de símbolos de BPSK, QPSK, 8PSK, QAM-16, QAM-32 y QAM-64. [20]

Un parámetro importante en el diseño de una modulación PSK y QAM es el número de símbolos utilizados. [20]

Una de las características de estas modulaciones es que en algunos casos se demuestra variación de fase, amplitud y frecuencia. Lo siguiente es una breve descripción de la Ecuación 1 para hallar los números de bits codificados y en la Tabla 1 se observa la comparativa de bits de acuerdo con cada modulación digital.

$$N = \text{Log}_2(M)$$

Donde:

N= Número de bits codificados

M= Cantidad de condiciones

Ecuación 1

Tabla 1 Comparativa de bits de modulaciones

Modulación	Símbolo (M)	unidades	Bits (N)
ASK, FSK, 2-PSK	2	bit	1
4-PSK, 4-QAM	4	Dibit	2
8-PSK, 8-QAM	8	Tribit	3
16-QAM	16	quadbit	4
32-QAM	32	Pentabit	5
64-QAM	64	Hexabit	6

Nota. Se puede observar los valores de bits codificados hallados mediante la

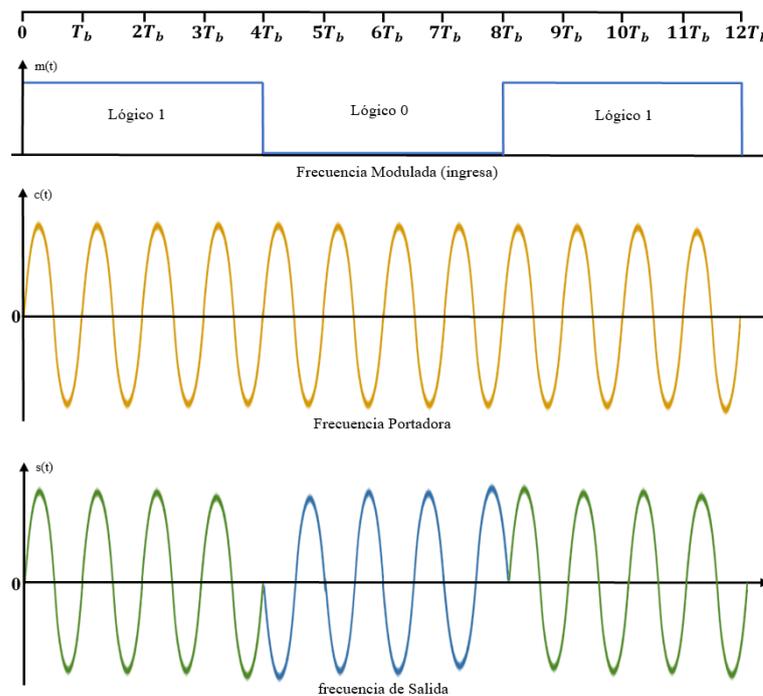
Ecuación 1 teniendo en cuenta que *M* son las posibles fases de salida para la señal portadora [21].

2.3 Modulación PSK (Phase Shift Keying)

Phase Shift Keying (PSK) es una modulación angular que implica cambiar la fase de la portadora. La señal modulada es una señal digital con un número finito de estados [22]. Este tipo de modulación se distingue por variar la fase de la señal portadora según la amplitud de la señal a modular, lo que resulta una modulación en fase. La señal portadora mantiene una amplitud constante y variaciones de fase entre 0° para un "0" lógico y 180° para un "1" lógico como se puede visualizar en la **Figura. 4**.

Las aplicaciones de la modulación PSK más comunes se encuentran en la televisión de alta definición para transmisión de señal HDTV, también en el espectro de potencia de la señal PSK es ampliamente utilizado por las empresas que brindan servicios de televisión por satélite porque cuanto mayor es la potencia, más pequeña es la antena parabólica.

Figura. 4 Modulación PSK

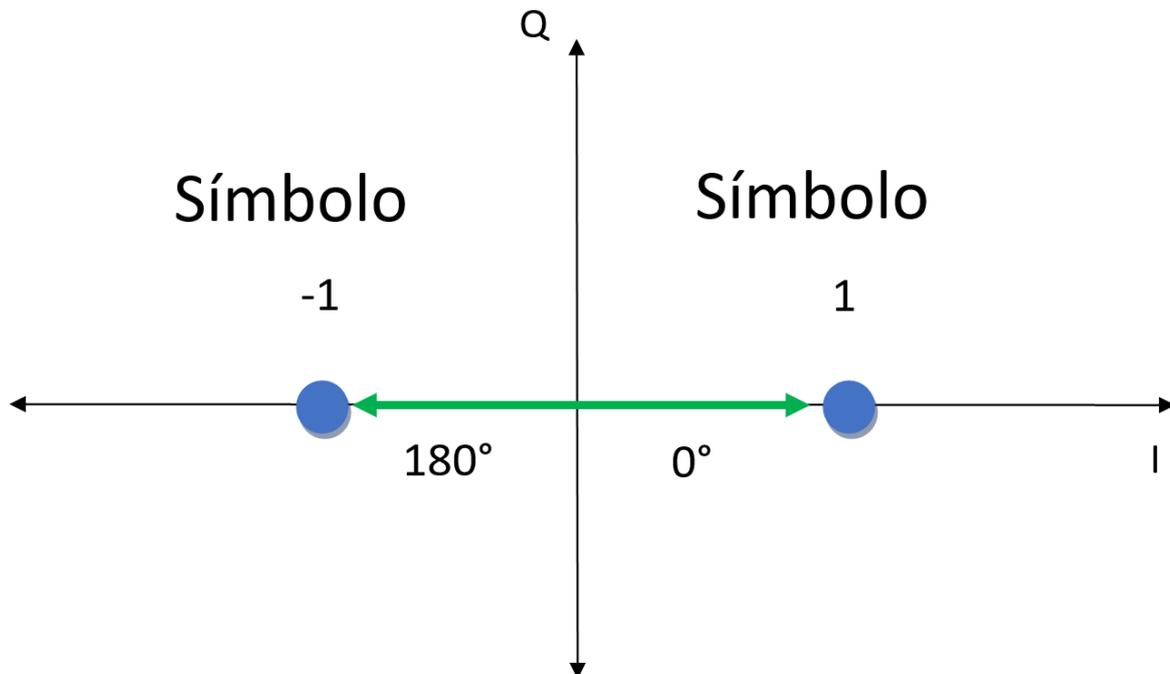


Nota. Se observa las variaciones de fase cada 4 tiempos basándose un "1" y "0" lógico en el ingreso de las señal modulada, y a su vez se aprecia como se fusiona la señal modulada con la portadora y da como resultado la señal de salida [22].

2.3.1 Modulación Por Desplazamiento De Fase En Binaria BPSK (2-PSK)

La modulación 2-PSK es una señal de doble banda lateral **Figura. 5**, esta modulación de onda cuadrada con supresión de portadora, donde frecuencias laterales están separadas de la frecuencia portadora por un valor igual a la mitad de la razón de bit en la cual tiene dos fases de salida y una sola frecuencia, las cuales representan 1 y 0 lógico respectivamente, es utilizada para transmisores que no requieren alta velocidad y bajo costo. [23]

Figura. 5 Modulación BPSK



Nota. Consta de 2 símbolos que se conforma de 1 Bit de información cada uno. [24]

2.3.2 Modulación Por Desplazamiento De Fase En Cuadratura QPSK (4-PSK)

En cada bit está encriptado dividido en parejas dándoles el valor nominal de la decoración como cambio de etapa en comparación con la señal anterior. La fase de la señal se modula para codificar bits de información digital en cada cambio de fase [23].

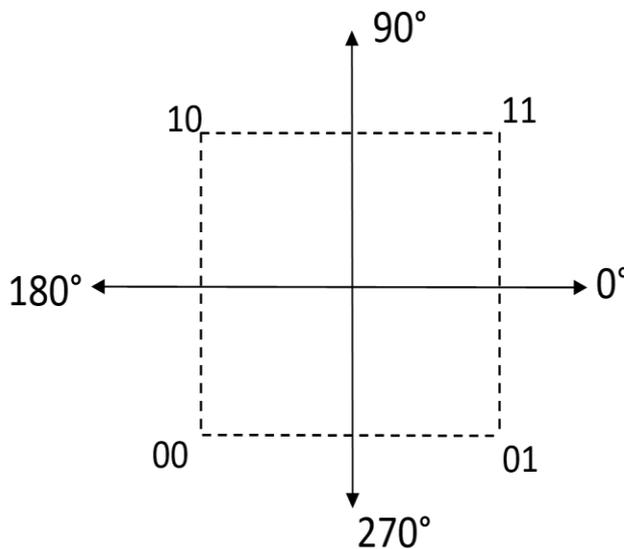
La fase de la señal portadora "N" puede tener un valor en serie distanciados en un ángulo

$\theta = \frac{2\pi}{N}$. En el caso $N = 4$, tiene 4-PSK o QPSK.

2.3.2.1 Codificación por cambio de fase en cuadratura

Codificación de compensación de fase cuadrática. En la **Figura. 6** representa las cuatro fases igualmente espaciadas: 00, 01, 11, 10, transmisión QPSK es de dos bits por fase el doble de eficiencia de ancho de banda Rango de frecuencia de BPSK. [25]

Figura. 6 Constelación QPSK



Los cuatro estados de salida tienen la misma amplitud y están separados 90°.

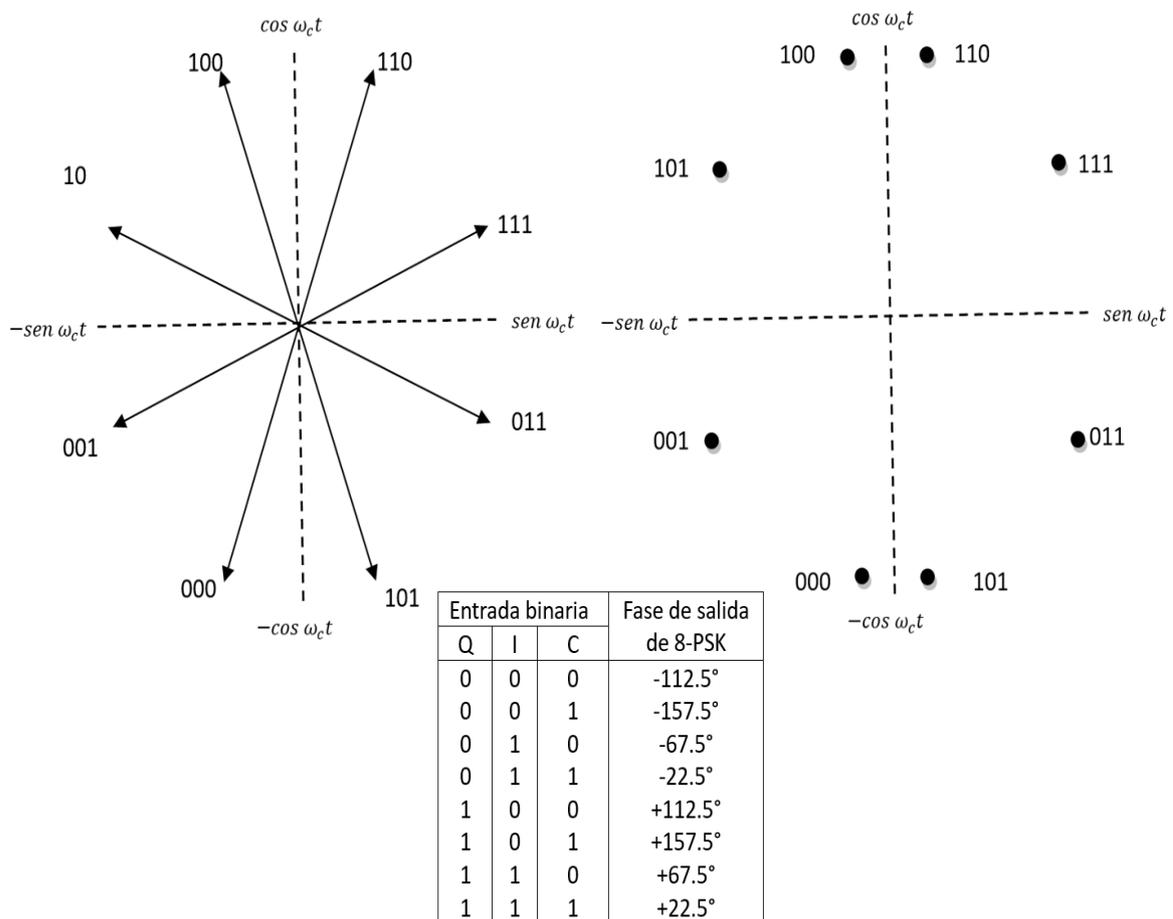
Nota. Estado de salida de la modulación QPSK. [25]

2.3.3 Modulación Por Desplazamiento De Fase En (8-PSK)

PSK de ocho fases (8 PSK) es una técnica para el símbolo donde $M = 8$. Con un adaptador de 8 PSK, hay ocho etapas de salida posibles. Para codificar ocho fases diferentes, los bits entrantes se consideran en grupos de 3 bits, llamados tribus ($2^3 = 8$).

Con 8-PSK, debido a que los datos se dividen en tres canales, la tasa de bits en el canal I, Q o C es un tercio de la tasa de datos de la entrada binaria $\left(\frac{f_b}{3}\right)$ ($f_b =$ velocidad de transmisión de bits sobre $N =$ número de bits) y (bits de cobertura ampliada I y Q) y C a tres veces su longitud en los bits de entrada). Dado que los bits I, Q y C tienen salidas simultáneas y paralelas, los convertidores de nivel 2 a 4 también ven crecer sus entradas (y por lo tanto su salida) a una tasa igual a $\left(\frac{f_b}{3}\right)$ como se muestra en la **Figura. 7**. [26]

Figura. 7 Constelación 8-PSK

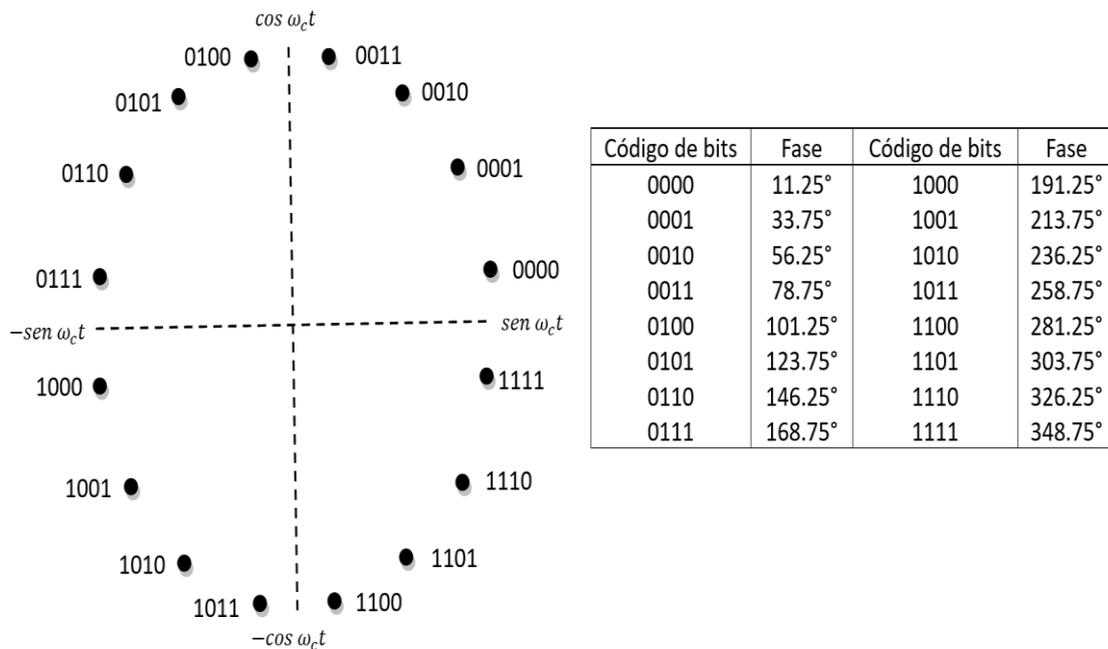


Nota. Entrada de bits y Salida de la modulación 8-PSK. [26]

2.3.4 Modulación Por Desplazamiento De Fase En (16-PSK)

El PSK de dieciséis etapas (16-PSK) es una técnica de codificación M-aria, donde $M = 16$; Hay 16 etapas de salida diferentes. El modulador 16-PSK opera con datos entrantes en grupos de 4 bits ($2^4 = 16$), llamados quads. La fase de salida no cambia hasta que se ingresan 4 bits en el modulador. Por lo tanto, la tasa de rotación de salida y el ancho de banda no son menos de una cuarta parte de la tasa de bits de entrada ($\frac{f_b}{4}$). La **Figura. 8** muestra la tabla de verdad y el diagrama de constelación del generador de 16 PSK. [26]

Figura. 8 Diagrama de constelación 16-PSK



Nota. Código de bits y fase de la modulación 16-PSK. [26]

Para una modulación 16-PSK, donde la separación entre fasores es de $\frac{\pi}{16}$ rad. El ancho de banda necesario para la modulación se reduce a la cuarta parte de la señal original, la pequeña separación entre fasores hace que este sistema sea vulnerable a errores en presencia de ruido porque el desvanecimiento del ruido puede causar desfasajes que se interpretarán como valores incorrectos en la recepción [19].

El estándar inalámbrico de ZigBee, también se basa en PSK. La norma IEEE 802.15.4 permite el uso de BPSK en 868 a 915 MHz y OQPSK en 2,4 GHz. La modulación 8-PSK es notablemente ausente de estos esquemas diversos. Esto se debe a que, aunque su tasa de error es similar a la de 16-QAM, su velocidad de datos es solo de tres cuartas partes de la de 16-QAM. Por lo tanto, 8-PSK rara vez se incluye en las normas y los esquemas tienden a "saltar" de QPSK a 16-QAM. Sin embargo, la modulación 8-QAM es posible usarla, pero hay que tener en cuenta que su implementación es difícil [27].

2.4 Modulación Qam (Quadrature Amplitude Modulation)

La modulación QAM es una modulación digital donde se modulará la información, tanto en amplitud (la amplitud no es constante) como en fase [28]. Esto significa que la señal portadora cambiará su amplitud y fase, juntos, para producir una salida analógica QAM.

QAM sería otro tipo de modulación M-aria utilizada grupos de K bits, se puede obtener $M = 2^k$ salidas diferentes.

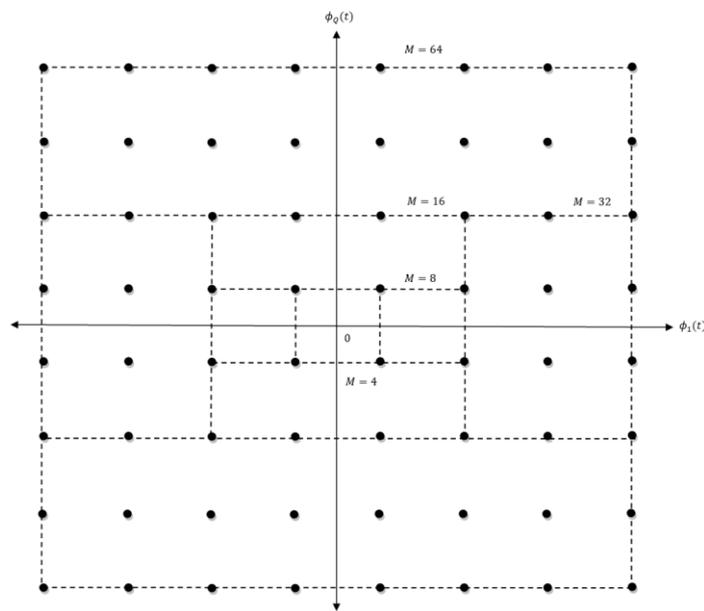
M-QAM es una modificación más general que incluye M-ASK y M-PSK como casos especiales. En QAM, los mensajes se codifican tanto en la amplitud como en la fase de la portadora. La idea básica detrás de QPSK explota el hecho de que $\cos(2\pi f_c t)$ y $\sin(2\pi f_c t)$ son perpendiculares en el período $[0, T_b]$ [29]. M-QAM son bidimensionales e incluyen dos funciones ortogonales básicas, dadas por las ecuaciones Ecuación 2 y Ecuación 3:

$$\phi_I(t) = \sqrt{\frac{2}{T_s}} \cos(2\pi f_c t), 0 \leq t \leq T_s \quad \text{Ecuación 2}$$

$$\phi_Q(t) = \sqrt{\frac{2}{T_s}} \sin(2\pi f_c t), 0 \leq t \leq T_s \quad \text{Ecuación 3}$$

La configuración más importante de los puntos de señal es el rectángulo. Los puntos de señal se conocen naturalmente como QAM rectangulares y se colocan en una cuadrícula rectangular espaciados uniformemente en anchura en cada dirección (dimensión). La **Figura. 9** muestra torres de señales QAM rectangulares de $M = 4, 8, 16, 32, 64$.

Figura. 9 Constelaciones QAM rectangulares



Nota. Constelaciones de 4, 8, 16, 32, 64-QAM rectangulares [29]

2.4.1 Aplicación De La Modulación QAM

La importancia de este sistema de modulación radica en su amplia gama de aplicaciones:

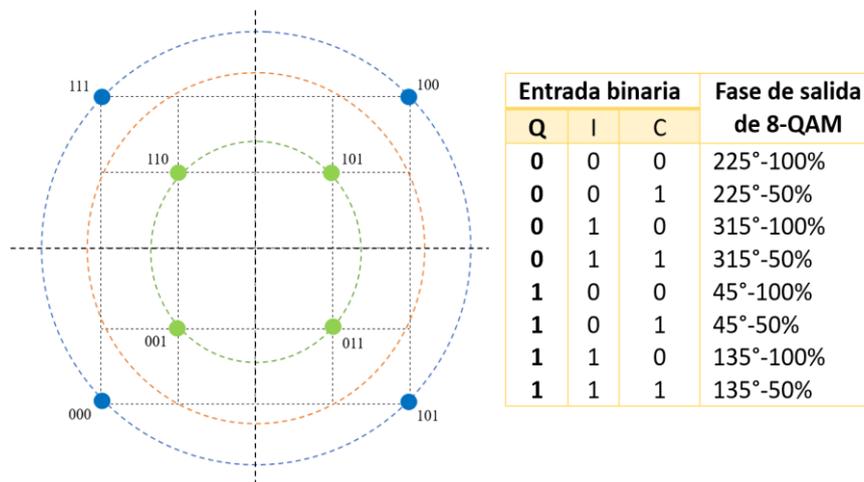
- Los módems que alcanzan velocidades superiores a 2400 bps, como V.22 bis y V.32, la utilizan. [30]
- Es la modulación utilizada en una variedad de sistemas de transmisión como la televisión, las microondas y las señales satélite. [30]
- Es la base de la modulación TCM (Trellis Coded Modulation), que combina la modulación con la codificación de canal para lograr velocidades de transmisión extremadamente altas. [30]
- Es el núcleo de los módems ADSL (Asymmetric Digital Subscriber Line) que funcionan en el ciclo de abonado y funcionan a frecuencias entre 24KHz y 1104KHz, Pueden alcanzar velocidades de hasta 9Mbps modulando en diferentes portadoras en QAM. [30]

2.4.2 Modulación De Amplitud En Cuadratura En (8-QAM)

La modulación de amplitud en cuadratura en (8-QAM), también conocido como QAM de ocho, es una técnica de codificación de M-ario, en la que $M = 8$. La señal de salida de un modulador de 8-QAM no es una señal de amplitud constante, a diferencia del 8-PSK [31].

Un sistema QAM 8-ario (8QAM) tiene ocho fasores, pero los divide en dos conjuntos de cuatro fasores cada uno con el mismo desfase y amplitud. De esta manera, se sigue reduciendo el ancho de banda mientras se mantiene un índice de sensibilidad al ruido aceptable.

Figura. 10 Modulación De 8-QAM



Nota. Constelación de 8-QAM rectangular con sus valores binario y fase de salida [19].

2.4.3 Modulación De Amplitud En Cuadratura En (16, 64, 256 -QAM)

En las modulaciones de amplitud en cuadratura en (16, 64, 256 -QAM), a medida que 16QAM pasa a 64QAM, etc., se pueden lograr velocidades de datos más altas, pero a expensas del margen de ruido.

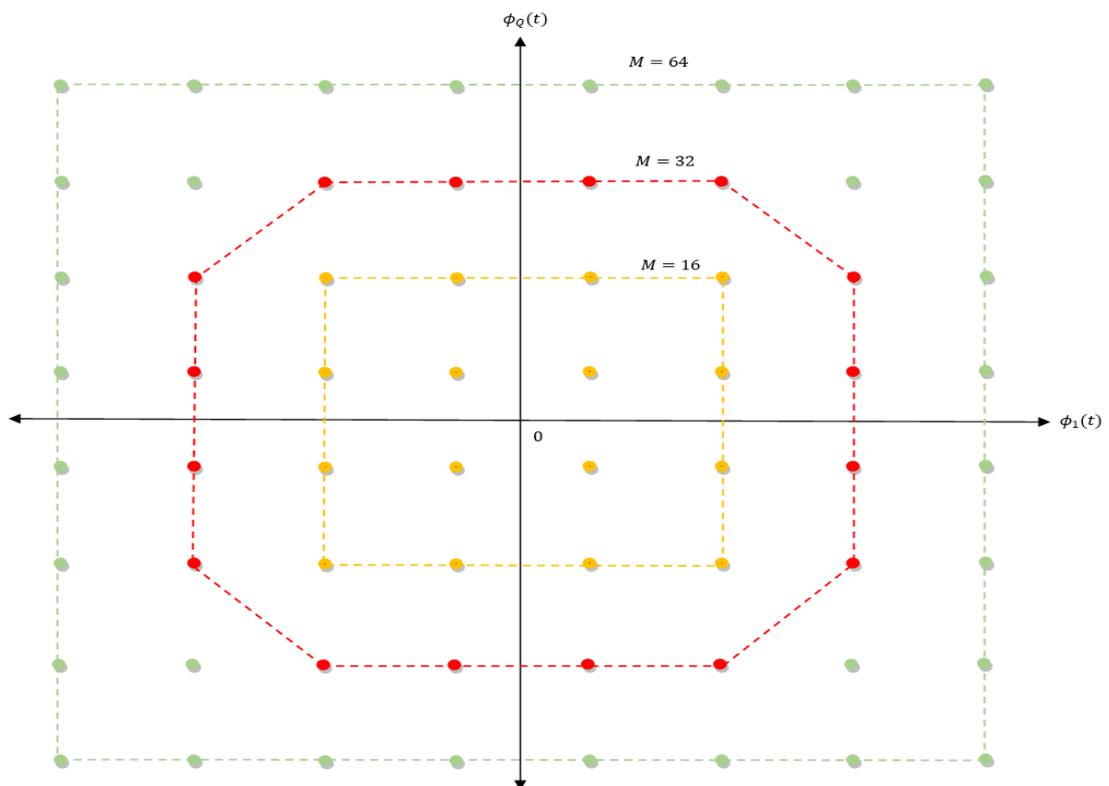
Muchos sistemas de transmisión de datos migran entre secuencias QAM, 16QAM, 32QAM, etc. diferente dependiendo del estado del enlace. Muchos sistemas de comunicaciones por radio que transmiten datos, como diversas formas de comunicaciones inalámbricas, comunicaciones celulares, etc., requieren este componente [32].

Existe un equilibrio entre la relación señal-ruido requerida y el rendimiento de datos. El rendimiento de datos que se puede lograr en condiciones ideales aumenta a medida que aumenta el orden de la señal QAM, por ejemplo, aumentando de 16QAM a 64QAM, etc.

Por lo tanto, la velocidad de datos se puede optimizar manteniendo la tasa de error requerida cambiando el orden de modulación y corrección de errores.

Las diferentes formas de modulación de amplitud en cuadratura (QAM) se muestran en diagramas de constelaciones, para diferentes formatos de modulación como se muestran en la **Figura. 11** en los siguientes diagramas:

Figura. 11 Constelaciones de 16, 32, 64-QAM



Nota. Se observa tres diagramas de constelaciones: 16-QAM en color amarillo, 32-QAM en color rojo y por último de 64-QAM que está en color verde.

2.5 Software Definido Por Radio (SDR)

El término "radio de software" fue acuñado en 1991 por Joe Mitola para referirse a una "radio" reconfigurable. Esto significa que el mismo hardware puede realizar diferentes funciones en diferentes momentos. De esta manera, demuestra el hecho de que existen dispositivos de hardware "universales" en la comunicación. En el mismo sentido, el concepto ha abierto con éxito la puerta a grandes oportunidades para la industria de la comunicación y la investigación. [33]

Con la llegada de los microprocesadores, los filtros digitales (DSP) e incluso los programas de computadora, todo este hardware puede funcionar sin tocar los controles de radio. La radio se vuelve parte de la computadora simplemente usando el software apropiado del fabricante de la radio y usando un puerto serial o paralelo. Además del administrador de dispositivos, se incluyen cientos de dispositivos de radio útiles. Desde vagones de telégrafo hasta sistemas de captura y transmisión de imágenes de televisión mediante tecnología de barrido lento. [33]

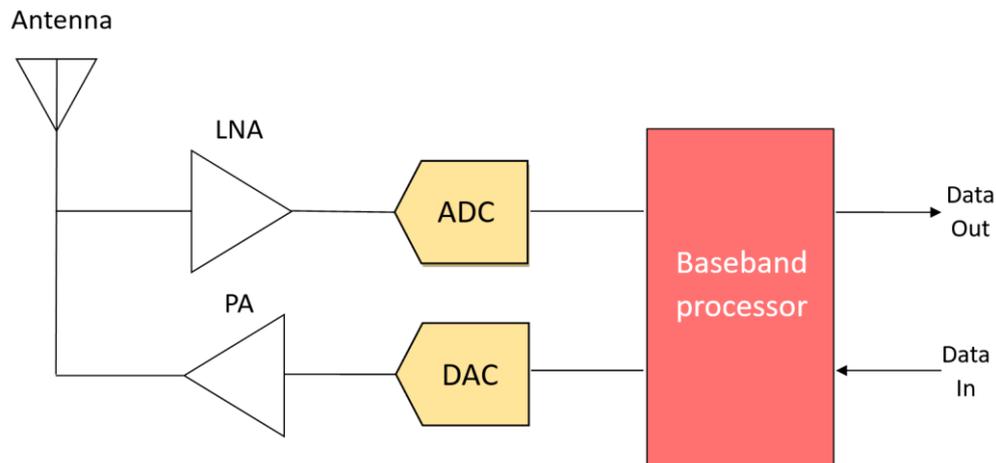
Sin embargo, los procesos que ingresan por la antena y salen por el amplificador, permanecen dentro de la radio, activando completamente los dispositivos, circuitos y componentes electrónicos físicos. El cambio fundamental que ha introducido SDR en el mundo de la radio es que simplifica enormemente la cantidad de materiales y circuitos que componen los equipos de radio y los reemplaza con software capaz de realizar funciones complejas, donde se necesitan demasiados circuitos específicos.

En la **Figura. 12** se puede ver la estructura del diseño típico de un sistema de comunicación por radio. El sistema consta de varios componentes interconectados que trabajan juntos para transmitir y recibir mensajes de forma inalámbrica.

Los principales componentes del sistema se pueden describir de la siguiente manera:

- **Antena:** es el componente que conecta el sistema con el medio de transmisión (en este caso el aire). Recibe y transmite señales de radiofrecuencia (RF) a través del conector.
- **LNA (Amplificador de Bajo Ruido):** Es un amplificador que se encarga de mejorar la señal de entrada del sistema, eliminando ruido y potenciando la señal. La señal mejorada se envía al siguiente componente: el PA (amplificador de potencia).
- **PA (amplificador de potencia):** este componente amplifica las señales de RF y las convierte en señales de mayor potencia necesarias para la transmisión. La señal amplificada se envía a una antena, donde se convierte en ondas de radio y se transmite por el aire.
- **ADC (Convertidor analógico a digital):** este componente convierte la señal analógica recibida a través de la antena en una señal digital que el sistema puede procesar. La señal digital se envía al procesador de banda base.
- **DAC (Convertidor de digital a analógico):** este componente realiza la función inversa de un ADC. Convierte la señal digital procesada por el procesador de banda base en una señal analógica y la envía a través de la antena.
- **Procesador de banda base:** este componente se encarga de procesar la señal digital recibida a través del ADC. Aquí se realiza la decodificación y codificación de datos y cualquier otra operación necesaria para transmitir o recibir información.
- **Entrada de datos:** los datos se ingresan al sistema utilizando fuentes externas como computadoras o dispositivos móviles.
- **Salida de datos:** después del procesamiento, los datos salen del sistema y se transfieren a una fuente externa, como otra computadora o dispositivo móvil.

Figura. 12 Sistema De Comunicación Ideal.



Nota. Comunicación Ideal Del SDR. [33]

SDR es una tecnología creada para mejorar la operabilidad entre diferentes servicios. La tecnología SDR incluye software y hardware que se pueden reconfigurar dinámicamente para comunicarse a través de una variedad de estándares de comunicación, protocolos y enlaces de radio. Los SDR se pueden usar para crear dispositivos de red e inalámbricos multifunción multibanda que se pueden reconfigurar dinámicamente, sujetos a actualizaciones de software y reconfiguración de hardware. [33]

Otra definición de SDR habla de "la medida en que parte o todo el hardware y la funcionalidad están definidos por el software", lo que significa que la arquitectura es flexible y configurable. La imagen es a veces en tiempo real, por lo que se puede modificar según parámetros, bandas de frecuencia, ancho de banda y patrones de reproducción. De esta manera, SDR es una radio multifuncional, programable y fácilmente actualizable que puede admitir una variedad de servicios y estándares y es una solución de bajo costo. [33]

La versatilidad de SDR está en una posición ideal para adaptarse a diferentes requisitos en lo que se conoce como Calidad de Servicio, QoS (Quality of Service), ya sea que requiera una gran cantidad de datos, voz, aplicaciones o multimedia. Actualmente,

algunas estaciones base están diseñadas para utilizar la arquitectura SDR o al menos alguna tecnología basada en SDR; sin embargo, las empresas de equipos están adoptando los principios de SDR en el desarrollo de radios multimodo y multi-estándar para el desarrollo de dispositivos pequeños, como teléfonos móviles y ordenadores. computadoras portátiles [33]

La radio definida por software es una tecnología de rápido desarrollo que goza de gran reconocimiento y apreciación en la industria de las telecomunicaciones. En los últimos años, los sistemas de radio analógicos han sido reemplazados por sistemas de radio digital, con aplicaciones de radio en los campos militar, civil y comercial. Además, los módulos de dispositivos programables han crecido significativamente, ya que comienzan a usarse en sistemas de radio digital en diferentes niveles de funcionalidad. La tecnología SDR tiene como objetivo explotar todos los módulos del hardware programable y así construir una arquitectura abierta basada en el sistema de radio.

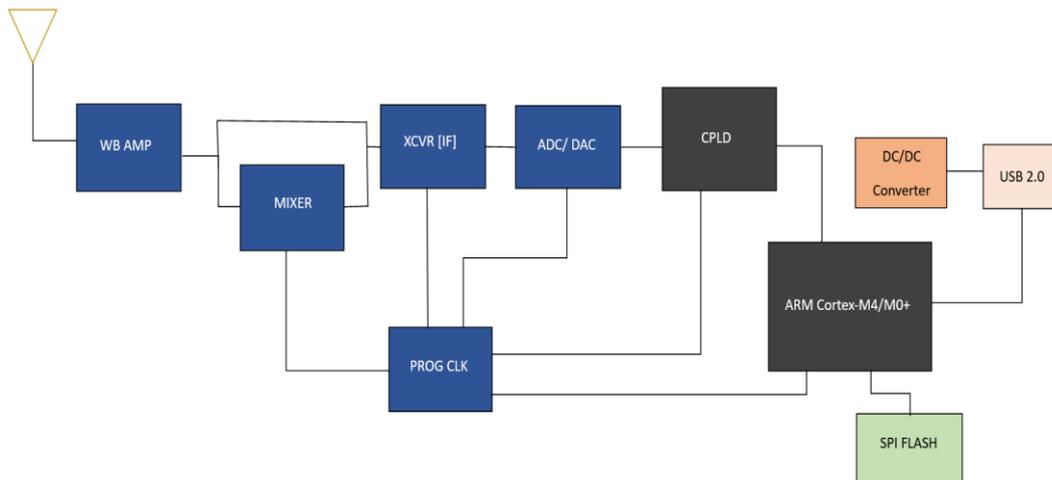
SDR se puede describir como un sistema de comunicación inalámbrico donde, Parte de los componentes son implementados por el programa en lugar de Implementación de hardware, uso para esto, asume un dispositivo integrado La información y su traslado a la computadora [34].

Este sistema permite Realice grandes cambios simplemente ajustando ciertos aspectos de Programación. Proporciona tecnologías de procesamiento de señales más allá del procesamiento de cintas banda ancha o estrecha y también crear funciones de seguridad para telecomunicaciones y Requisitos de forma de onda. Ingeniería de hardware convencional del sistema. Radio, compuesta principalmente por detectores, filtros, mezcladores, moduladores, Amplificador o demodulador, reemplazado en SDR con software instalado En un PC u otro dispositivo basado en microcontrolador [35].

2.5.1 Esquema Genérico Del SDR

En la explicación de los bloques que conforman el diagrama electrónico de la **Figura. 13** será detallado para que el lector pueda comprender el funcionamiento principal de cada bloque del sistema electrónico del SDR:

Figura. 13 Diagrama genérico de los bloques electrónicos que componen un SDR



Nota. Funcionamiento general de los equipos SDR [35]

WB AMPL

Es un amplificador de RF de banda ancha que sirve como última etapa de amplificación. Está implementado tanto en la opción de receptor como en la de transmisor, por lo que en el diagrama aparecerán como dos amplificadores separados, cada uno en dirección opuesta. [35]

MIXER O MEZCLADOR

Se trata de un mezclador, utilizado tanto en la recepción como en transmisión para poder manejar señales fuera de la banda de 2,15 a 2,75 GHz (en este caso se establecerá directamente la conexión al último bloque. Como se puede apreciar en el esquema)de la **Figura. 13**, la señal recibida fuera de la banda anterior se puede conmutar a 2,15 y 2,75 GHz utilizando la portadora local generada en el mezclador; Usando el proceso inverso del

caso de transmisión, transmita esta señal de 2.15 a 2.75 GHz a cualquier otra frecuencia entre 1 MHz y 6 GHz, usando en una transmisión. [35]

XCVR [IF]

Es un transceptor que puede recibir o transmitir señales hacia o desde la banda de 2.15 a 2.75 GHz directamente desde la banda base, por lo que en esta aplicación se usa como una frecuencia de Convertidor Intermedio (IF) para luego cambiar a la frecuencia final usando el mezclador y siempre que la frecuencia objetivo no esté en el rango indicado anteriormente, el uso de esta frecuencia no será superfluo en este caso. Por tanto, se puede decir que la aplicación utiliza una arquitectura de doble turno de forma general, aunque se opta por la conmutación directa entre 2,15 y 2,75 GHz. [35]

ADC/DAC

Este es un convertidor dúplex que consta de ADC (convertidor analógico a digital) y DAC (convertidor digital a analógico), también tiene capacidad de dúplex para que ambos convertidores puedan funcionar simultáneamente. [35]

CPLD

Es un dispositivo lógico programable complejo que se utiliza como interfaz entre el convertidor ADC/DAC y el procesador del dispositivo para actuar como un puente lógico entre los protocolos de recepción de datos de dos bloques. [35]

ARM Cortex-M4/M0+

Es un microcontrolador de doble núcleo con arquitectura ARM de la familia Cortex-M, que presenta el consumo más bajo de la gama Cortex, y Cortex-M0 es el núcleo de menor consumo entre ARM. [35]

PROG CLK

Es una integración que nos permite generar simultáneamente varios relojes con una frecuencia independiente para cada nuevo reloj, a partir de un solo cristal como referencia primaria. [35]

SPI FLASH

Es una memoria flash conectada a un microcontrolador ARM, que se puede reconfigurar y nos permite configurar el microcontrolador maestro y el CPLD, cada vez que se inicia el HackRF One. [35]

DC/DC Converter

Se trata de un convertidor de caída de tensión (topología buck) con dos salidas de diferente tensión que constituye la fuente de alimentación de la placa y se alimenta de la tensión de entrada del puerto USB 2.0. [35]

USB 2.0

Esta es la interfaz encargada de alimentar el HackRF One y permitir la transferencia de datos entre nuestro ordenador y la placa de circuito en los modos de funcionamiento transmisor y receptor. [35]

A continuación, se muestra en la **Figura. 14** la placa diseño completo de HackRF One:

Figura. 14 Placa Impresa del HackRF One



Nota. Placa Del Hackrf One [35]

2.5.2 Características Del Hackrf One

2.5.2.1 Antena Telescópica ANT 500

Este tipo de antena es muy simple y consta de varillas conductoras dispuestas verticalmente. Se utiliza principalmente en la banda de alta frecuencia (onda corta) y banda de frecuencia de modulación (FM), por lo que su tamaño no es muy grande. Esta antena está montada en el exterior de la radio. Es un dipolo con una impedancia característica de 50Ω que opera en el rango de frecuencia de 75 MHz a 1 GHz, con una longitud total configurable de 20 a 88 cm. Bueno para trabajar con HackRf One. [3]

La **Figura. 15** muestra la antena telescópica que se está utilizando, con una longitud máxima de 88 cm.

Figura. 15 Antena ANT500.



Nota. Antena Telescópica Ant500 Con Conector SMA RF [36]

Lóbulo de la antena

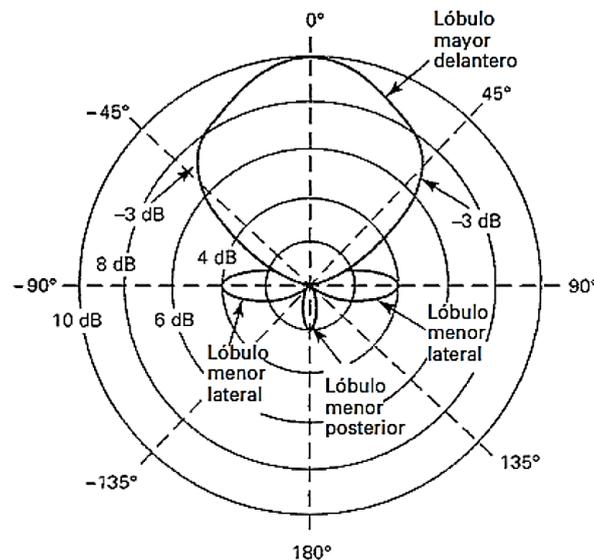
El lóbulo principal de una antena es el rango angular de direcciones sobre las cuales el patrón de radiación del haz está 3 dB por debajo de su valor máximo. Es decir, la potencia radiada disminuye a la mitad [37]

En un patrón de radiación de antena, la intensidad de campo o densidad de potencia se representa mediante diferentes posiciones angulares con respecto a la antena.

Puedes identificar uno o más lóbulos primarios que reciben la mayor cantidad de energía y algunos lóbulos secundarios o laterales que representan radiación no deseada. Puede haber lóbulos dorsales o laterales. La relación de potencia entre el lóbulo frontal y el lóbulo posterior se denomina directividad, y la relación entre el lóbulo frontal y el lóbulo lateral se denomina relación lateral frontal [38]

En la **Figura. 16**, se pueden identificar los diferentes lóbulos de una antena no especificada en un diagrama de radiación relativa.

Figura. 16 lóbulos de radiación



Nota. patrón de radiación de la antena ANT500. [39]

2.5.3 El Concepto de Arquitectura de SDR.

La arquitectura SDR es versátil y flexible y utiliza hardware de uso general que se puede programar o configurar mediante software. En comparación con las arquitecturas tradicionales que usan muestreo en cuadratura, SDR muestrea usando una frecuencia intermedia, que generalmente procesa más la señal en el dominio digital. [40]

En comparación con las soluciones tradicionales, las ventajas de SDR provienen de su fácil adaptación al entorno y la cantidad de aplicaciones compatibles. Esto se debe a que el software se puede desacoplar por completo de las operaciones de hardware mediante una capa intermedia llamada Capa de abstracción de hardware (HAL). El propósito de esta capa es asegurar la portabilidad y reutilización del software. Estos pequeños marcos de propósito general permiten que el desarrollo de aplicaciones sea independiente del hardware de radio. [40]

El concepto SDR se basa en antenas de RF y procesamiento de señal digital (DSP). En el caso de las antenas, esto se refleja en la flexibilidad de sintonizar distintas bandas de frecuencia, beamforming (adaptativa y autocancelable), operación MIMO (multiple input multiple output) y supresión de interferencias. Obviamente, es más fácil apuntar a todos estos parámetros de antena en teoría que en la práctica. [40]

En términos de procesamiento de señales analógicas y digitales, SDR maneja diferentes niveles de RF, lo que lo hace más adaptable a entornos cambiantes para lograr los niveles de QoS (Calidad de servicio) requeridos. Además, esta flexibilidad se extiende a los módulos de conversión de señal mixta de datos configurados para recibir la señal deseada. Por ejemplo, el procesamiento de señales digitales también se configura y programa para complementar el filtro de RF en términos de realizar su función. [40]

2.5.4 Arquitectura de Rf para SDR

Los sistemas de comunicación de última generación buscan la optimización de los dispositivos, integrando un mayor control del software y la capacidad de enviar o recibir detalles específicos, y realizar sus tareas de manera eficiente sin obstaculizar a los usuarios que requieren un diseño de RF que sea flexible en términos de frecuencia y ancho de banda; Condiciones que disfruta la radio portátil (walkie-talkie). La arquitectura de los sistemas superheterodinos con una o dos fases de mezcla es una parte integral del diseño

de radio [41]. Ejemplos del desarrollo de dichos sistemas son los dispositivos de microondas y radiofrecuencia con un rendimiento mejorado al reducir el consumo de energía. Por el momento, reducir el consumo de energía es una característica importante de una forma de vida respetuosa con el planeta.

Los convertidores de analógico a digital (ADC) y los convertidores de digital a analógico (DAC) tienen tasas de muestreo más altas, linealidad y un recuento de bits efectivo. El poder de procesamiento de los FPGA permite algoritmos más eficientes, integrados y calibrados digitalmente. Esta tecnología redujo la densidad de la batería del dispositivo y mejoró la gestión térmica; sin embargo, la optimización del dispositivo genera rendimientos decrecientes. Si bien los componentes de RF continúan disminuyendo en tamaño, peso y potencia, los filtros de alto rendimiento siguen siendo físicamente grandes y, a menudo, tienen un diseño personalizado, lo que aumenta el costo total del sistema. Los filtros MF producen un ancho de banda de canal analógico básico que dificulta la creación de un diseño de plataforma común que pueda reutilizarse en una amplia variedad de sistemas. [41]

En esta revisión se menciona la búsqueda de alternativas (Zero-IF Architecture (ZIF)), que es una opción posible; Es un receptor de etapa de mezcla de frecuencia única con un oscilador local configurado directamente en la banda de frecuencia respectiva, y traduce la señal recibida en la banda base en una señal de fase (I) y cuadratura (Q). Esta nueva arquitectura permite el filtrado de banda base, con filtros que son fáciles de diseñar y acceder.

Esta frecuencia de cambio de banda base directa puede conducir a la pérdida de componentes de frecuencia y portadora de la imagen. Debido a factores del mundo real en la cadena de la señal, no se puede mantener una diferencia de fase ideal de 90° . La diferencia de fase entre las señales I y Q da como resultado una imagen degradada. [41]

Los sistemas de próxima generación requieren la integración de una arquitectura ZIF en un transceptor monolítico, con cascadas analógicas y de RF en una sola pieza de silicio. La variación del proceso se reducirá y los bloques DSP (procesamiento de señal digital) se pueden integrar en el transceptor eliminando el límite entre el algoritmo de calibración en cuadratura y la cadena de señal.

Finalmente, el Adalm Pluto SDR integra toda la cadena de señal de RF, analógica y digital en un solo dispositivo e incluye DSP aplicado para la corrección en tiempo real de la portadora y la dispersión del puente. Método A través de conversiones de proceso, frecuencia y temperatura, Zero-IF integra ingeniería general . El AD9361 está dirigido a un rendimiento medio con baja potencia, como enlaces de datos, sistemas de comunicación móvil y aplicaciones SDR. [41]

2.6 Universal Software Radio Peripheral (USRP)

USRP es un dispositivo que permite la creación de SDR utilizando cualquier computadora con un puerto USB 2.0 o USB 3.0 así como se muestra en la **Figura. 17**. Consiste en módulos de hardware con funciones de transmisión y recepción en un amplio rango de frecuencias. [42]

Figura. 17 Equipo USRP



Nota. Usrp B210 Marca Ni - National Instruments. [43]

2.6.1 Comparación de SDR Vs Hackrf One

Cuando se trata de comparar equipos SDR (Software Defined Radio) como se visualiza en la **Tabla 2** se tiene los siguientes: RTL-SDR, HackRF One y USRP B210, es importante considerar sus características y capacidades para determinar qué opción se adapta mejor a las necesidades del proyecto.

En comparación con el RTL-SDR, el HackRF One ofrece mayor flexibilidad y capacidades avanzadas, ya que tiene la capacidad de transmisión y recepción le permite realizar experimentos más complejos, modulación y análisis de señales más complejos.

Los USRP B210 (Universal Software Radio Peripherals) son ahora un dispositivo de hardware popular para experimentar con comunicaciones en tiempo real usando SDR. Teniendo en cuenta que ahora hay dispositivo como HackRF One que tiene un rango de frecuencia operativa de 1-6 MHz, lo que permite que múltiples HackRF se conecten y trabajen juntos.

El HackRF One está en un rango intermedio entre el RTL-SDR y el USRP B210 en términos de precio. Ofrece un buen equilibrio entre prestaciones y costo, lo que puede ser atractivo para aquellos que buscan capacidades más avanzadas sin llegar al costo del USRP B210.

Tabla 2 Comparación entre RTL-SDR, HackRF One y USRP B210

Equipos SDR	RTL.SDR	HackRF One	USRP B210
Rango De Frecuencia	500khz – 1.766 Mhz	1 Mhz – 6 GHz	50 Mhz – 6 GHz

Transmisión Compatible	Receptor	Half Duplex	Full Duplex
Ancho De Banda RF	2.4 Mhz	20 Mhz	61.44 Mhz
Sample Size (ADC/DAC)	8 bits	8 bits	12 bits
Interfaz	USB 2.0 (480 megabit)	USB 2.0 (480 megabit)	USB 3.0 (5 gigabit)
Precio	\$15 - \$25	\$300	\$1100

El dispositivo SDR que es compatible, con este proyecto es el HackRF One porque brinda una buena relación calidad-precio y cubre con todas las necesidades del proyecto ya que, tiene una amplia gama de funcionalidades, el HackRF One es capaz de trabajar con modulaciones PSK (Phase Shift Keying) y QAM (Quadrature Amplitude Modulation), teniendo una variedad de frecuencias que va desde 1Mhz a 6Ghz en la cual, se puede transmitir y recibir señales, lo que lo hace adecuado con este proyecto que involucran estas técnicas de modulación.

CAPÍTULO 3: Metodología

Para elaborar este proyecto se usa el tipo metodología de investigación aplicada a la tecnológica, este se basa en conocimientos que puedan ponerse en práctica y se enfoque en ocasionar un impacto positivo, con el fin de ir desglosando cada parte que compone una modulación, también se encuentra la investigación cualitativa y cuantitativa, donde se realizó el análisis de tesis, revistas, artículos y demás información que pudiera dar el conocimiento general y técnico para la implementación de modulaciones digitales con tecnología SDR.

A continuación, en la **Figura. 18** se observa la conexión de los equipos:

Figura. 18 Conexión del módulo



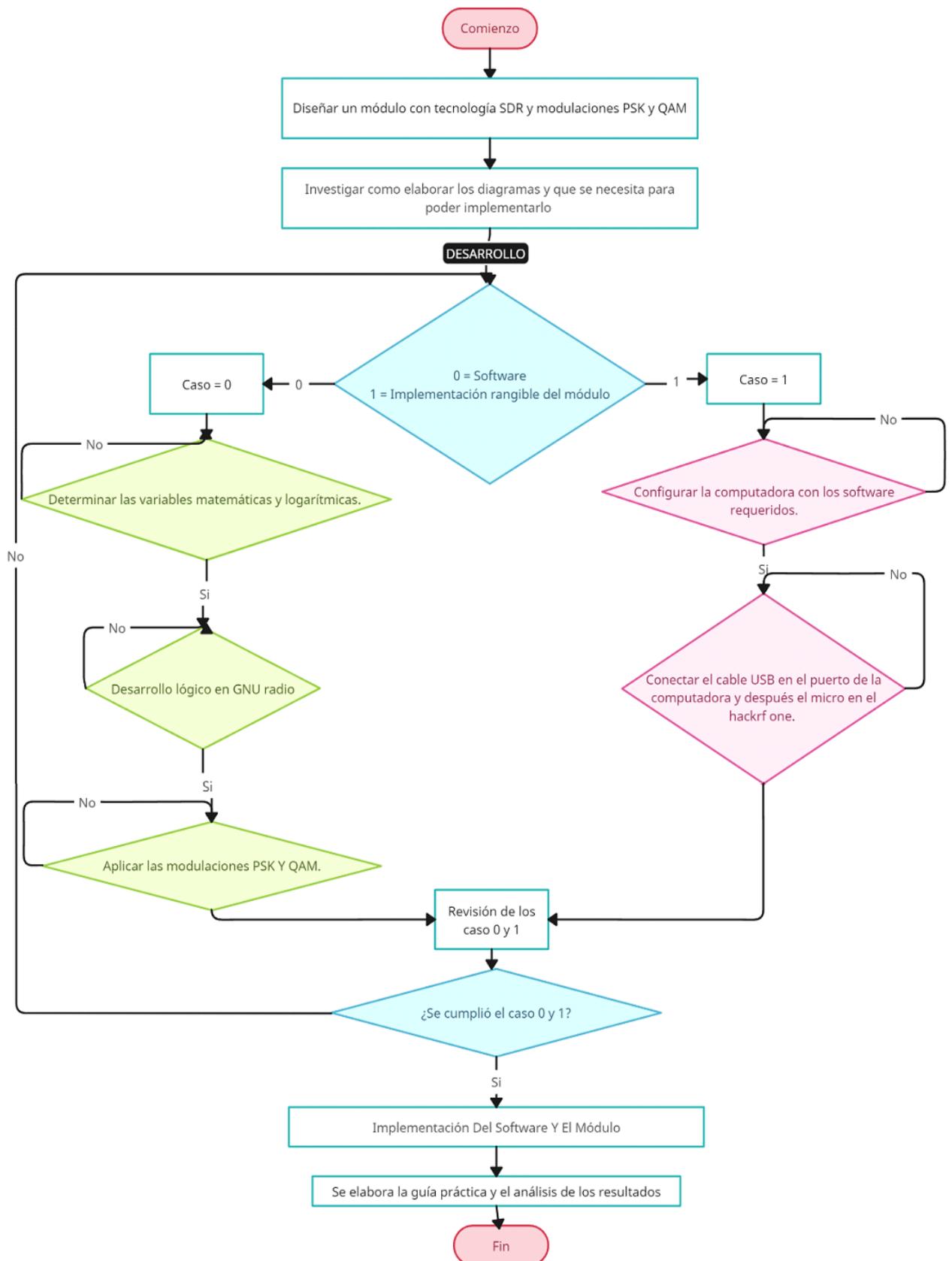
Nota. Elaborador por el autor.

DIAGRAMA DEL DESARROLLO DE LA PROPUESTA

En la propuesta de este proyecto se presenta una estructura metódica y detallada que articula los diferentes aspectos y pasos necesarios para llevar a cabo esta investigación, que está enfocada en el proceso de enseñanza de los estudiantes en las asignaturas “Comunicaciones Digitales y Procesamiento Digital de Señales”. Además, puede explorar la efectividad de diversas técnicas de enseñanza para mejorar la comprensión y el desempeño de los estudiantes en las asignaturas como: comunicaciones digitales y el procesamiento de señales digitales.

Este diagrama del desarrollo de la propuesta de la **Figura. 19**, está diseñado con el objetivo de proporcionar una visión general y estructurada de la propuesta, permitiendo detallar de manera clara y visual la secuencia lógica, y los componentes esenciales que conforman el proceso de investigación ya que, cada fase se encuentra interconectada, y respalda el progreso continuo hacia la culminación del estudio, facilitando la planificación, ejecución y seguimiento de cada fase, lo que asegura la coherencia y la adecuada realización de la investigación propuesta.

Figura. 19 Diagrama de la propuesta



Nota. Elaborador por el autor.

3.1 Hardware Para SDR

HackRF One que se muestra en la **Figura. 20** es un dispositivo de radio definido por software capaz de transmitir o recibir Señales de radio de 1MHz a 6GHz fabricadas por Great Scott Gadgets.

Figura. 20 Equipo HackRF One



Nota. El Hackrf One Consta Conectores Sma, USB 2.0. [3]

Fue Diseñado para facilitar el desarrollo y prueba de tecnología (testing and hacking) Sistema de radiocomunicaciones existente (soporta LTE) y en desarrollo para Generaciones de tecnologías de radio y sus protocolos. HackRF One es un archivo Plataforma de hardware libre que se puede usar como periférico a través de USB o programado para trabajar de forma independiente. A continuación, se detallan las principales características del dispositivo, así como Varias habilidades le dan una gran fuerza y flexibilidad:

Tabla 3 Características del HackRF One

Característica	Valor
Rango de frecuencias	1MHz-6GHz
Ancho de banda RF	20MHz

Profundidad de muestreo	8 bits
Velocidad de muestreo	20MSPS
Canales del transmisor	1
Receptores	1
Dúplex	Semidúplex
Interfaz	USB 2.0
Puertas lógicas programables	USB 2.0
Chipset	MAX5864, MAX2837, RFFC5072
Código abierto	Código abierto completo
Potencia de transmisión	-10dBm+ (15dBm @ 2.4GHz)
Compatible	Windows, Linux, incluidos SDR# y GNU Radio.

Nota. El Hackrf One Tienes estas características [44] .

3.1.1 Medios computacionales necesarios en el proceso SDR

El diseño de un SDR proporciona ciertos recursos computacionales necesarios para implementar la aplicación más compleja del caso. Recursos informáticos que incluyen GPP (procesador de uso general), DSP (procesador de señal digital), FPGA (matriz de puertas programables en campo) y, a veces, otros chips que amplían las capacidades informáticas. [40]

Un ejemplo de GPP que toman muchos desarrolladores de SDR es PowerPC, que es una clase de procesadores programados en lenguaje C o C++. Esta clase de procesadores realiza más de 1 billón de operaciones por segundo. Otra característica de estos GPP es la ejecución de instrucciones en paralelo, a menudo realizando tareas como evaluaciones lógicas, cálculos aritméticos y decisiones de bifurcación. [40]

Más importante que los procesos de modulación y demodulación de la señal, es la velocidad a la que se realizan estos procesos. El procesamiento de señales representa más del 90 % de la carga computacional, pero los protocolos involucrados en estas redes suelen representar el 90 % de las líneas de código. Por lo tanto, es muy importante que el hardware y la arquitectura SDR estén diseñados incluyendo hardware para el procesamiento de señales digitales, de modo que la señal se procese a la velocidad más alta y los procesadores GPP no se sobrecarguen con sus funciones. [40]

Un procesador de señal digital (DSP) es algo diferente de un GPP. Un DSP o Digital Signal Processor es un microprocesador diseñado específicamente para el procesamiento de señales digitales. Las características más básicas, como la forma de la aritmética, la velocidad, la organización de la memoria y la arquitectura interna, determinan la idoneidad para una aplicación en particular. Amplia disponibilidad de herramientas de desarrollo. [45]

Estrictamente hablando, el término DSP significa cualquier chip que opera con señales representadas digitalmente. En la práctica, el término se refiere a los microprocesadores diseñados específicamente para realizar el procesamiento de señales digitales. Los DSP utilizan arquitecturas especiales para acelerar los cálculos matemáticos intensivos inherentes a la mayoría de los sistemas de procesamiento de señales en tiempo real. Por ejemplo, las arquitecturas DSP incluyen circuitos para realizar rápidamente operaciones de multiplicación y acumulación denominadas MAC. Una arquitectura de memoria que permite múltiples accesos, a menudo de manera que

múltiples operandos, como muestras de señales de entrada o coeficientes de filtro, pueden cargarse simultáneamente en periféricos especiales o interfaces de entrada/salida que permiten que el procesador cargue múltiples operandos. Se comunica de manera efectiva con el resto de los componentes del sistema, como los convertidores A/D y la memoria. [45]

El uso de procesadores FPGA significa que el procesamiento de la señal no está definido por los lenguajes de programación tradicionales, por ejemplo, C, sino que utiliza lenguajes como VHDL, que están diseñados para arquitecturas de hardware específicas. La señal de radio está descrita en Circuitos Integrados de Muy Alta Velocidad (VHSIC) y Lenguaje de Diseño de Hardware (VHDL), aunque esto la hace portátil, no sigue una secuencia de instrucciones, por lo que el software no se desarrolla automáticamente de la forma habitual, por lo que la implementación de FPGA es mucho más costosa. [40]

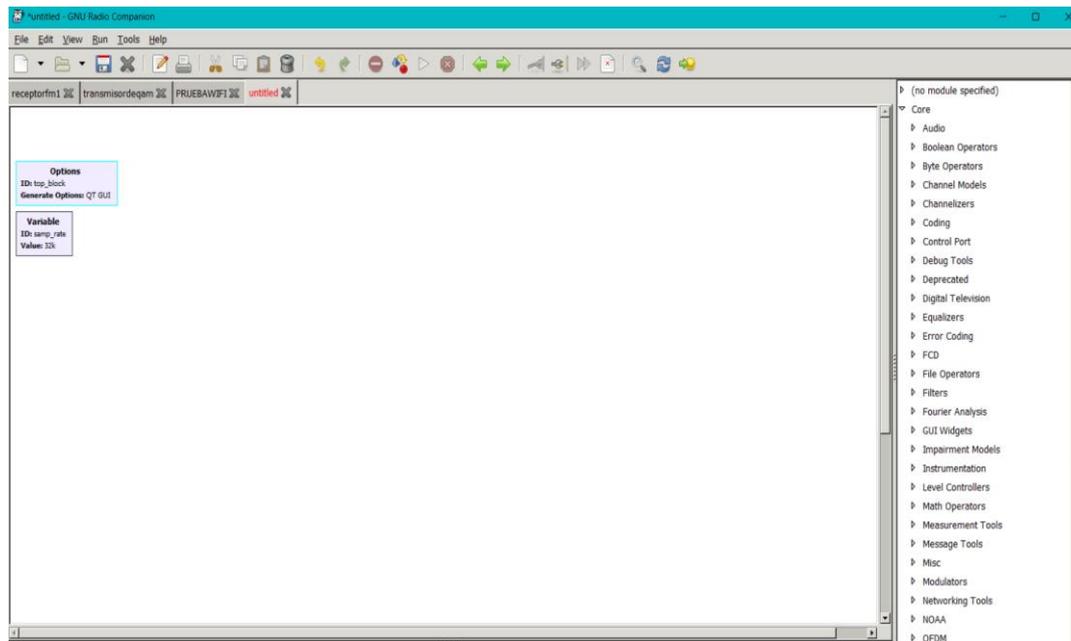
Actualmente, SDR ofrece la combinación correcta de alternativas vistas anteriormente, GPP, DSP, FPGA, lo que garantiza que se pueda implementar una amplia gama de aplicaciones con un nivel aceptable de recursos. [40]

3.2 Software Para SDR

3.2.1 Gnu Radio

Gnu Radio es un programa de código abierto diseñado para crear bloques a fin de implementar sistemas SDR, y cuenta con un entorno de simulación totalmente especializado para recrear ambientes óptimos en futuras comunicaciones inalámbricas como se observa en la **Figura. 21**. [46]

Figura. 21 Vista de trabajo en Gnu Radio

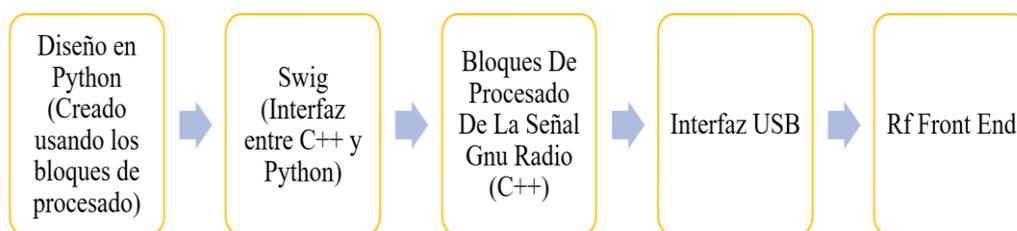


Nota. Ventana de configuración de los bloques.

Dirigida a entusiastas de la radio, entornos educativos y comerciales, Gnu Radio está destinada a ayudar o asistir en la investigación de sistemas de comunicación inalámbricos y de radio ubicados en cualquier parte del mundo.

A continuación, la arquitectura de Radio Gnu se muestra en la **Figura. 22**.

Figura. 22 Arquitectura Gnu Radio



Nota. Diagrama de los bloques de la arquitectura de Gnu Radio [47]

A continuación, se explicará la configuración del hardware HackRF One usando el sistema operativo Windows 11 y el software Gnu Radio V3.7.11, configurando los bloques requeridos para modulaciones QAM y PSK.

Para comprender los esquemas o bloques de implementación, es necesario considerar que no existen módulos con cadenas de conexión, Las variables generadas por el usuario o herramienta para organizar el ploteo de resultados.

Por lo tanto, un aspecto importante son los datos que se utilizarán para cada módulo del Gnu Radio, dado que cada módulo tiene una entrada y una salida de datos, dependiendo del comportamiento que se puede cambiar, por ejemplo, cuando se carga y coloca el bit de módulo, en la salida que se tiene una cadena.

3.3 Instalación Gnu Radio

GNU Radio Companion (GRC) es una interfaz gráfica que facilita el diseño y la implementación de flujos de trabajo, mientras que Gqrx es una aplicación de radio definida por software (SDR) que le permite convertir, recibir y explorar señales de radio con dispositivos compatibles.

Esta guía está diseñada para proporcionar instrucciones sobre cómo instalar en el sistema operativo Windows 10 Home: GNU Radio Companion (GRC) y el software Gqrx.

Nota: En la **Figura. 23** se encuentra la carpeta con los instaladores de Gnu Radio:https://upse-my.sharepoint.com/:f:/g/personal/ruby_saenzvalero_upse_edu_ec/Eq-D0ryrrHdHrVB6PWT_6PEBCJIOHtsii-MteuS3Z1Qag?e=kyF9CT

Figura. 23 Carpeta de instaladores en OneDrive

Implementación De Sistemas Definidos Por Radio Basado En Código Abierto Para Modulaciones PSK Y QAM En Fase De Cuadratura						
Nombre	Modificado	Modificado por	Tamaño de arch...	Compartir	Actividad	
 GNURadio Components MWK.7z	Hace unos segundos	Saenz, Ruby	452 MB	 Compartido		

Nota. Elaborador por el autor.

Una vez descargada la carpeta **GNURadio Componentets MWK.7z**, se procede a abrir la carpeta y se va al instalador donde dice **vc_redist.x86.exe** así como se observa en la **Figura. 24** y da clic en ejecutar.

Figura. 24 *vc_redist.x86.exe*

Carpeta de archivos	
How to Setup.7z	2.144 ? Archivo WinRAR 11/4/2020 10:40 9F7E69A8
Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	178.915 445.296.327 Python Wheel 8/11/2018 2:57 D2C730FE
get-pip.py	1.807.342 ? Python File 9/4/2020 6:53 B6090F5A
gnuradio_3.7.11_win64.msi	340.172.800 ? Paquete de Windo... 9/4/2020 9:29 5CDE06B1
gtk2-runtime-2.24.10-2012-10-10-ash.exe	7.944.501 28.562.386 Aplicación 4/11/2018 19:01 74F1FDD7
Join My Channel in youtube.txt	221 ? Documento de texto 11/4/2020 10:41 189BA733
lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl	3.074.020 ? Python Wheel 8/11/2018 2:56 D7ED920E
numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	28.691.052 ? Python Wheel 8/11/2018 2:57 421E8549
pygtk-2.22.0-cp27-none-win_amd64.whl	963.000 ? Python Wheel 4/11/2018 18:44 41D398C4
PyOpenGL-3.1.0-py2-none-any.whl	1.746.912 ? Python Wheel 8/11/2018 2:59 B6B05520
PyQt4-4.11.4-cp27-cp27m-win_amd64.whl	49.968.470 ? Python Wheel 4/11/2018 17:57 4362F953
PyQt5-5.2.1.win-amd64.ReleaseDLL.exe	587.656 ? Aplicación 8/11/2018 3:31 CDF40A0B
python-2.7.15.amd64.msi	20.246.528 ? Paquete de Windo... 8/11/2018 3:20 20C54925
uhd_3.9.6-release_x64_VS2015.exe	8.024.000 ? Aplicación 4/11/2018 16:49 CFBF048E
vc_redist.x86.exe	14.500.216 ? Aplicación 8/11/2018 3:20 DF70FDBB
wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl	11.293.430 ? Python Wheel 8/11/2018 2:59 E04AE083
wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe	593.230 ? Aplicación 8/11/2018 3:35 E7BB48D7

Nota. Elaborador por el autor.

Instalar **Python-2.7.15.amd64.msi** como esta en la **Figura. 25** , tener en cuenta que se debe establecer para todos los usuarios.

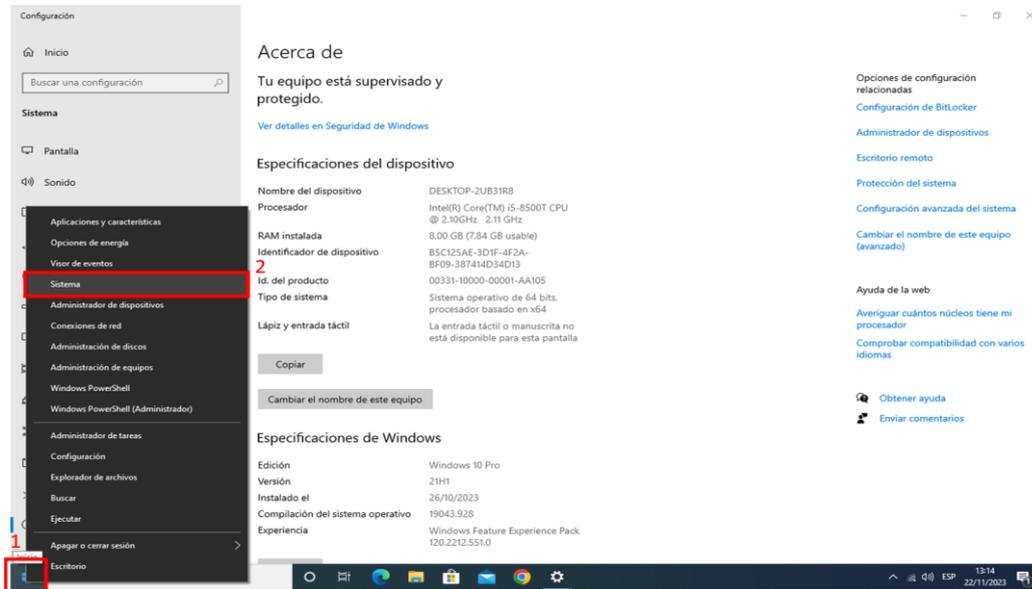
Figura. 25 *Python-2.7.15.amd64.msi.*

Carpeta de archivos	
How to Setup.7z	2.144 ? Archivo WinRAR 11/4/2020 10:40 9F7E69A8
Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	178.915 445.296.327 Python Wheel 8/11/2018 2:57 D2C730FE
get-pip.py	1.807.342 ? Python File 9/4/2020 6:53 B6090F5A
gnuradio_3.7.11_win64.msi	340.172.800 ? Paquete de Windo... 9/4/2020 9:29 5CDE06B1
gtk2-runtime-2.24.10-2012-10-10-ash.exe	7.944.501 28.562.386 Aplicación 4/11/2018 19:01 74F1FDD7
Join My Channel in youtube.txt	221 ? Documento de texto 11/4/2020 10:41 189BA733
lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl	3.074.020 ? Python Wheel 8/11/2018 2:56 D7ED920E
numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	28.691.052 ? Python Wheel 8/11/2018 2:57 421E8549
pygtk-2.22.0-cp27-none-win_amd64.whl	963.000 ? Python Wheel 4/11/2018 18:44 41D398C4
PyOpenGL-3.1.0-py2-none-any.whl	1.746.912 ? Python Wheel 8/11/2018 2:59 B6B05520
PyQt4-4.11.4-cp27-cp27m-win_amd64.whl	49.968.470 ? Python Wheel 4/11/2018 17:57 4362F953
PyQt5-5.2.1.win-amd64.ReleaseDLL.exe	587.656 ? Aplicación 8/11/2018 3:31 CDF40A0B
python-2.7.15.amd64.msi	20.246.528 ? Paquete de Windo... 8/11/2018 3:20 20C54925
uhd_3.9.6-release_x64_VS2015.exe	8.024.000 ? Aplicación 4/11/2018 16:49 CFBF048E
vc_redist.x86.exe	14.500.216 ? Aplicación 8/11/2018 3:20 DF70FDBB
wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl	11.293.430 ? Python Wheel 8/11/2018 2:59 E04AE083
wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe	593.230 ? Aplicación 8/11/2018 3:35 E7BB48D7

Nota. Elaborador por el autor.

Para verificar las características de la computadora se hace clic derecho en el **ícono Windows**, luego haga clic en **Sistema** como en la **Figura. 26**.

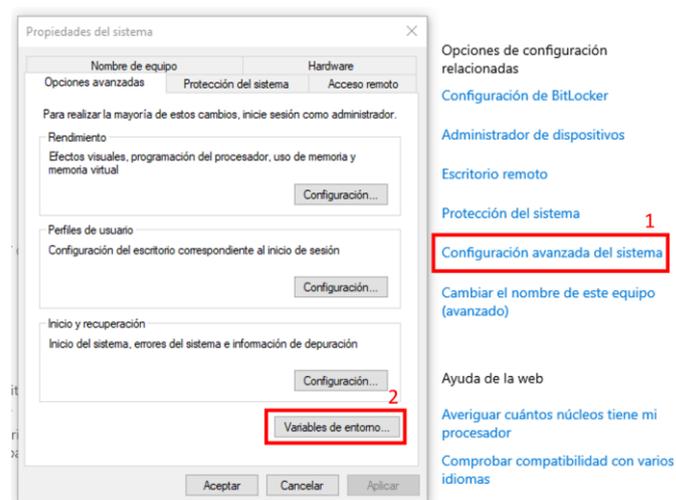
Figura. 26 Pantalla del Sistema



Nota. Elaborador por el autor.

Haga clic en **Configuración avanzada del sistema**, luego de clic en **Variable de entorno** como se muestra en la **Figura. 27**.

Figura. 27 Configuración avanzada del sistema y Variable de entorno .

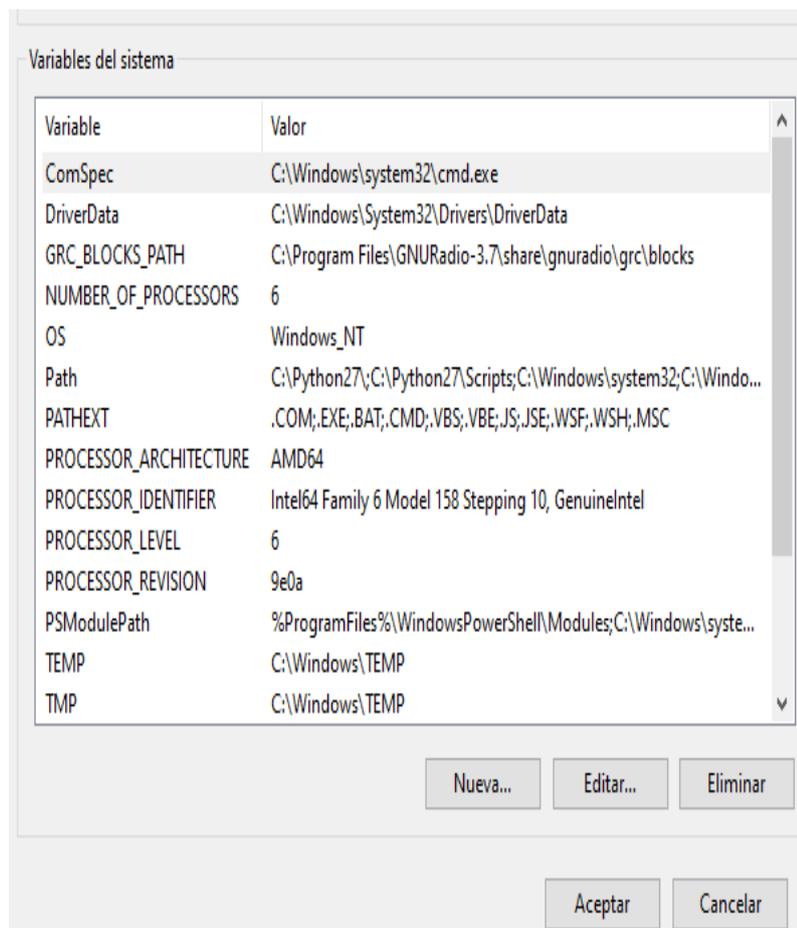


Nota. Elaborador por el autor.

En el cuadro de **variables del sistema**, en **Path** dar clic y verifique la ruta de Python si se agregó como en la **Figura. 28**. Si no, debe incluir de esta manera: Copie la ubicación de Python. Luego, agregue su ubicación de esta manera:

- Abre C:\Python27 y agrega otra:
- scripts en C:\Python27
- Luego haga clic en Aceptar.

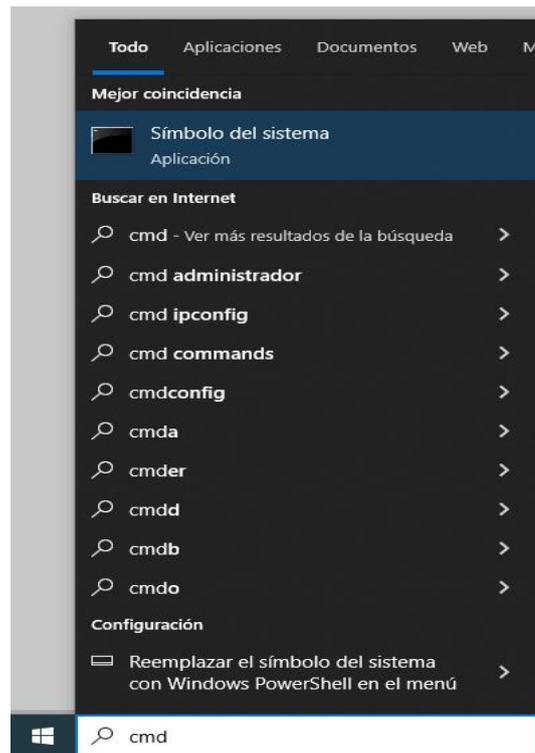
Figura. 28 Variables del sistema.



Nota. Elaborador por el autor.

Abra el **símbolo del sistema (CMD)** en la ventana con inicio como en la **Figura. 29**.

Figura. 29 Pantalla de símbolo del sistema (CMD)



Nota. Elaborador por el autor.

Verifique primero si Python funciona, escribiendo en la terminal: Python como se observa en la **Figura. 30**.

Figura. 30 CMD Python.

A screenshot of a Windows Command Prompt window titled 'Símbolo del sistema - python'. The window shows the following text:

```
Microsoft Windows [Versión 10.0.19043.928]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\pc>python
Python 2.7.17 (v2.7.17:c2f86d86e6, Oct 19 2019, 21:01:17) [MSC v.1500 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Nota. Elaborador por el autor.

Instalar el archivo que se muestra en la **Figura. 31** que esta con el nombre de **Python get-pip.py**, Ahora espere hasta que se complete la instalación.

Figura. 31 Python get-pip.py.

..			Carpeta de archivos		
How to Setup.7z	2.144		? Archivo WinRAR	11/4/2020 10:40	9F7E69A8
Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	178.915	445.296.327	Python Wheel	8/11/2018 2:57	D2C730FE
get-pip.py	1.807.342		? Python File	9/4/2020 6:53	B6090F5A
gnuradio_3.7.11_win64.msi	340.172.800		? Paquete de Windo...	9/4/2020 9:29	5CDE06B1
gtk2-runtime-2.24.10-2012-10-10-ash.exe	7.944.501	28.562.386	Aplicación	4/11/2018 19:01	74F1FDD7
Join My Channel in youtube.txt	221		? Documento de texto	11/4/2020 10:41	189BA733
lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl	3.074.020		? Python Wheel	8/11/2018 2:56	D7ED920E
numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	28.691.052		? Python Wheel	8/11/2018 2:57	421E8549
pygtk-2.22.0-cp27-none-win_amd64.whl	963.000		? Python Wheel	4/11/2018 18:44	41D398C4
PyOpenGL-3.1.0-py2-none-any.whl	1.746.912		? Python Wheel	8/11/2018 2:59	B6B05520
PyQt4-4.11.4-cp27-cp27m-win_amd64.whl	49.968.470		? Python Wheel	4/11/2018 17:57	4362F953
PyQwt-5.2.1.win-amd64.ReleaseDLL.exe	587.656		? Aplicación	8/11/2018 3:31	CDF40A0B
python-2.7.15.amd64.msi	20.246.528		? Paquete de Windo...	8/11/2018 3:20	20C54925
uhd_3.9.6-release_x64_VS2015.exe	8.024.000		? Aplicación	4/11/2018 16:49	CFBE048E
vc_redist.x86.exe	14.500.216		? Aplicación	8/11/2018 3:20	DF70FDBB
wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl	11.293.430		? Python Wheel	8/11/2018 2:59	E04AE083
wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe	593.230		? Aplicación	8/11/2018 3:35	E7B848D7

Nota. Elaborador por el autor.

Instalar todos los archivos (.whl) que se muestran en la **Figura. 32**, debe instalarlo cambiando el directorio en cmd a la carpeta de componentes de Gnu radio.

- lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl

Después de terminar la instalación, continúe con estos archivos:

- numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl
- PyQt4-4.11.4-cp27-cp27m-win_amd64.whl
- PyQwt-5.2.1.win-amd64.ReleaseDLL.exe (Instalación normal)
- Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl
- PyOpenGL-3.1.0-py2-none-any.whl
- pygtk-2.22.0-cp27-none-win_amd64.whl
- gtk2-runtime-2.24.10-2012-10-10-ash.exe (Instalación normal)
- wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl

– wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe (Instalación normal)

Figura. 32 Archivos (.whl)

		Carpeta de archivos			
How to Setup.7z	2.144	?	Archivo WinRAR	11/4/2020 10:40	9F7E69A8
Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	178.915	445.296.327	Python Wheel	8/11/2018 2:57	D2C730FE
get-pip.py	1.807.342		Python File	9/4/2020 6:53	B6090F5A
gnuradio_3.7.11_win64.msi	340.172.800		Paquete de Windo...	9/4/2020 9:29	5CDE06B1
gtk2-runtime-2.24.10-2012-10-10-ash.exe	7.944.501	28.562.386	Aplicación	4/11/2018 19:01	74F1FDD7
Join My Channel in youtube.txt	221		Documento de texto	11/4/2020 10:41	189BA733
lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl	3.074.020		Python Wheel	8/11/2018 2:56	D7ED920E
numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	28.691.052		Python Wheel	8/11/2018 2:57	421E8549
pygtk-2.22.0-cp27-none-win_amd64.whl	963.000		Python Wheel	4/11/2018 18:44	41D398C4
PyOpenGL-3.1.0-py2-none-any.whl	1.746.912		Python Wheel	8/11/2018 2:59	B6B05520
PyQt4-4.11.4-cp27-cp27m-win_amd64.whl	49.968.470		Python Wheel	4/11/2018 17:57	4362F953
PyQt5-5.2.1.win-amd64.ReleaseDLL.exe	587.656		Aplicación	8/11/2018 3:31	CDF40A0B
python-2.7.15.amd64.msi	20.246.528		Paquete de Windo...	8/11/2018 3:20	20C54925
uhd_3.9.6-release_x64_VS2015.exe	8.024.000		Aplicación	4/11/2018 16:49	CFBE048E
vc_redist.x86.exe	14.500.216		Aplicación	8/11/2018 3:20	DF70FDBB
wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl	11.293.430		Python Wheel	8/11/2018 2:59	E04AE083
wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe	593.230		Aplicación	8/11/2018 3:35	E7BB48D7

Nota. Elaborador por el autor.

Instalar el archivo que esta seleccionado en el **Figura. 33** con el nombre: **uhd_3.9.6-release_x64_VS2015.exe**, dando clic derecho en ejecutar, tener en cuenta que se debe establecer para todos los usuarios.

Figura. 33 uhd_3.9.6-release_x64_VS2015.exe.

		Carpeta de archivos			
How to Setup.7z	2.144	?	Archivo WinRAR	11/4/2020 10:40	9F7E69A8
Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	178.915	445.296.327	Python Wheel	8/11/2018 2:57	D2C730FE
get-pip.py	1.807.342		Python File	9/4/2020 6:53	B6090F5A
gnuradio_3.7.11_win64.msi	340.172.800		Paquete de Windo...	9/4/2020 9:29	5CDE06B1
gtk2-runtime-2.24.10-2012-10-10-ash.exe	7.944.501	28.562.386	Aplicación	4/11/2018 19:01	74F1FDD7
Join My Channel in youtube.txt	221		Documento de texto	11/4/2020 10:41	189BA733
lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl	3.074.020		Python Wheel	8/11/2018 2:56	D7ED920E
numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	28.691.052		Python Wheel	8/11/2018 2:57	421E8549
pygtk-2.22.0-cp27-none-win_amd64.whl	963.000		Python Wheel	4/11/2018 18:44	41D398C4
PyOpenGL-3.1.0-py2-none-any.whl	1.746.912		Python Wheel	8/11/2018 2:59	B6B05520
PyQt4-4.11.4-cp27-cp27m-win_amd64.whl	49.968.470		Python Wheel	4/11/2018 17:57	4362F953
PyQt5-5.2.1.win-amd64.ReleaseDLL.exe	587.656		Aplicación	8/11/2018 3:31	CDF40A0B
python-2.7.15.amd64.msi	20.246.528		Paquete de Windo...	8/11/2018 3:20	20C54925
uhd_3.9.6-release_x64_VS2015.exe	8.024.000		Aplicación	4/11/2018 16:49	CFBE048E
vc_redist.x86.exe	14.500.216		Aplicación	8/11/2018 3:20	DF70FDBB
wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl	11.293.430		Python Wheel	8/11/2018 2:59	E04AE083
wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe	593.230		Aplicación	8/11/2018 3:35	E7BB48D7

Nota. Elaborador por el autor.

Instalar GNU Radio como se observa en la esta con el nombre: **gnuradio_3.7.11_win64.msi** e instalarlo para todos los usuarios ahora, después de finalizar la instalación

Figura. 34 Instalador de Gnu Radio.

		Carpeta de archivos				
..						
How to Setup.7z	2.144	?	Archivo WinRAR	11/4/2020 10:40	9F7E69A8	
Cheetah-2.4.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	178.915	445.296.327	Python Wheel	8/11/2018 2:57	D2C730FE	
get-pip.py	1.807.342		?	Python File	9/4/2020 6:53	B6090F5A
gnuradio_3.7.11_win64.msi	340.172.800		?	Paquete de Windo...	9/4/2020 9:29	5CDE06B1
gtk2-runtime-2.24.10-2012-10-10-ash.exe	7.944.501	28.562.386		Aplicación	4/11/2018 19:01	74F1FDD7
Join My Channel in youtube.txt	221		?	Documento de texto	11/4/2020 10:41	189BA733
lxml-3.5.0-cp27-cp27m-win_amd64.Release.whl	3.074.020		?	Python Wheel	8/11/2018 2:56	D7ED920E
numpy-1.10.4-cp27-cp27m-win_amd64.ReleaseDLL.whl	28.691.052		?	Python Wheel	8/11/2018 2:57	421E8549
pygtk-2.22.0-cp27-none-win_amd64.whl	963.000		?	Python Wheel	4/11/2018 18:44	41D398C4
PyOpenGL-3.1.0-py2-none-any.whl	1.746.912		?	Python Wheel	8/11/2018 2:59	B6B05520
PyQt4-4.11.4-cp27-cp27m-win_amd64.whl	49.968.470		?	Python Wheel	4/11/2018 17:57	4362F953
PyQt-5.2.1.win-amd64.ReleaseDLL.exe	587.656		?	Aplicación	8/11/2018 3:31	CDF40A0B
python-2.7.15.amd64.msi	20.246.528		?	Paquete de Windo...	8/11/2018 3:20	20C54925
uhd_3.9.6-release_x64_VS2015.exe	8.024.000		?	Aplicación	4/11/2018 16:49	CFBE048E
vc_redist.x86.exe	14.500.216		?	Aplicación	8/11/2018 3:20	DF70FDBB
wx-3.0-cp27-none-win_amd64.ReleaseDLL.whl	11.293.430		?	Python Wheel	8/11/2018 2:59	E04AE083
wxPython-common-3.0.2.0.win-amd64.ReleaseDLL.exe	593.230		?	Aplicación	8/11/2018 3:35	E7BB48D7

Nota. Elaborador por el autor.

Una vez que se instaló Gnu radio, en el escritorio te va a aparecer el icono de la aplicación como en la **Figura. 35**.

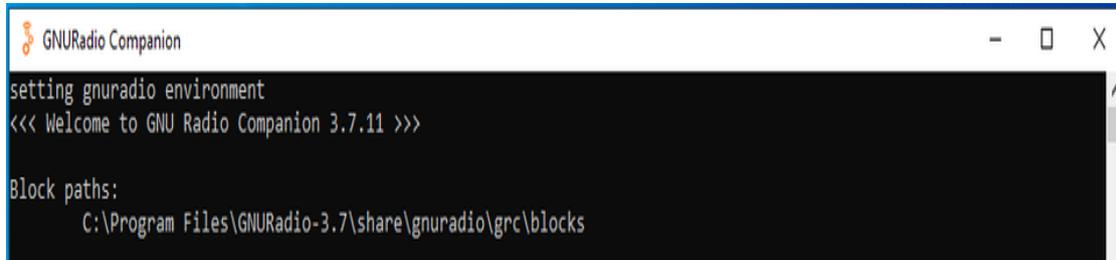
Figura. 35 Aplicación Gnu Radio Companion.



Nota. Elaborador por el autor.

Hacer doble clic en el icono de Gnu Radio, donde va a compilar la configuración del entorno de Gnu Radio (Consola) y nos muestra la ventana que está en la **Figura. 36**.

Figura. 36 Configuración del entorno de Gnu Radio.



```

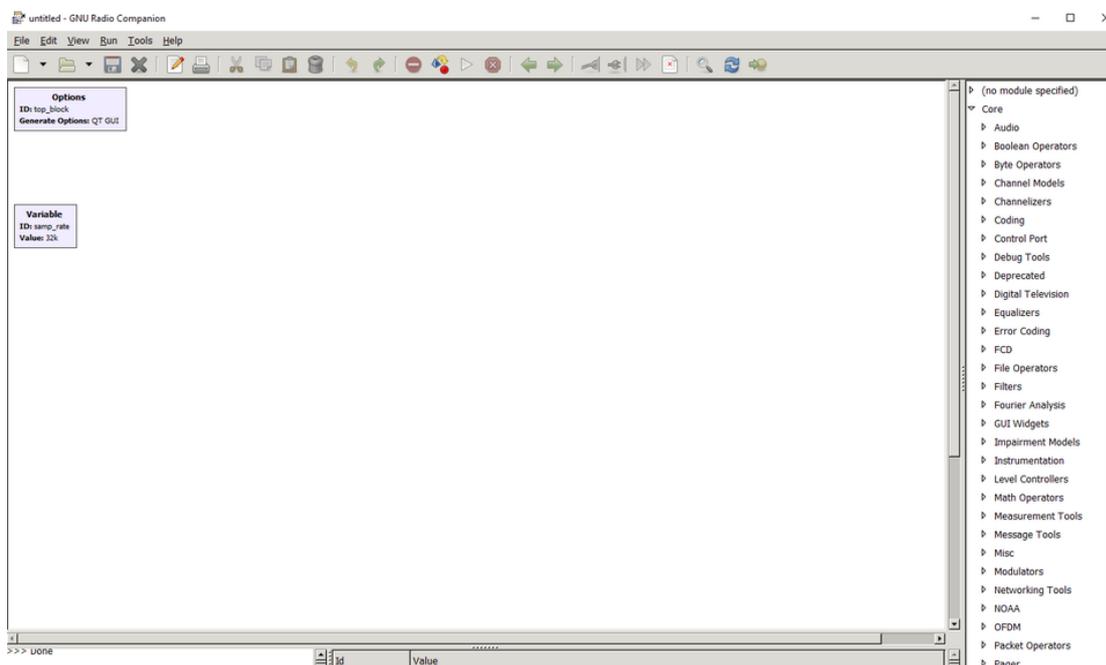
GNURadio Companion
setting gnuradio environment
<<< Welcome to GNU Radio Companion 3.7.11 >>>

Block paths:
  C:\Program Files\GNURadio-3.7\share\gnuradio\grc\blocks
  
```

Nota. Elaborador por el autor.

En la **Figura. 37** se observa el entorno gráfico donde se van a configurar los bloques.

Figura. 37 Entorno gráfico de Gnu Radio.



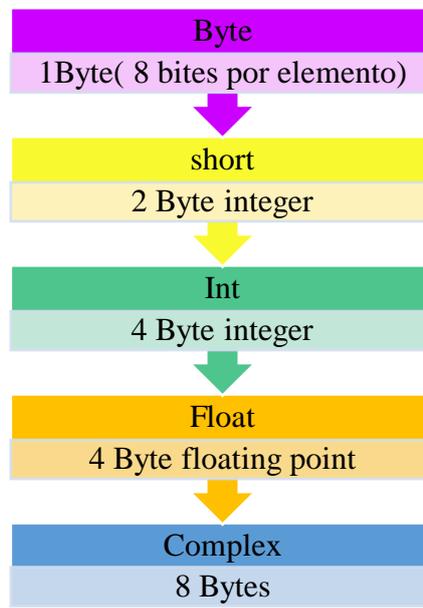
Nota. Elaborador por el autor.

3.4 Tipos De Datos En Gnu Radio

Cada tipo de datos está designado por un color diferente, así como se observa en la

Tabla 4:

Tabla 4 Tipos de datos en Gnu Radio.



Nota. Elaborador por el autor.

3.5 Materiales

Los materiales seleccionados en la **Tabla 5**, tendrán un impacto directo en la ejecución de esta propuesta. Cada material será crucial para el desarrollo del proyecto, desde componentes físicos (hardware) hasta recursos digitales (software) en **Anexo 3** y **Anexo 4**.

Tabla 5 Materiales

MATERIALES	
HARDWARE	SOFTWARE
Computadora	Gnu Radio V.3.11.7
HackRf One	Matlab

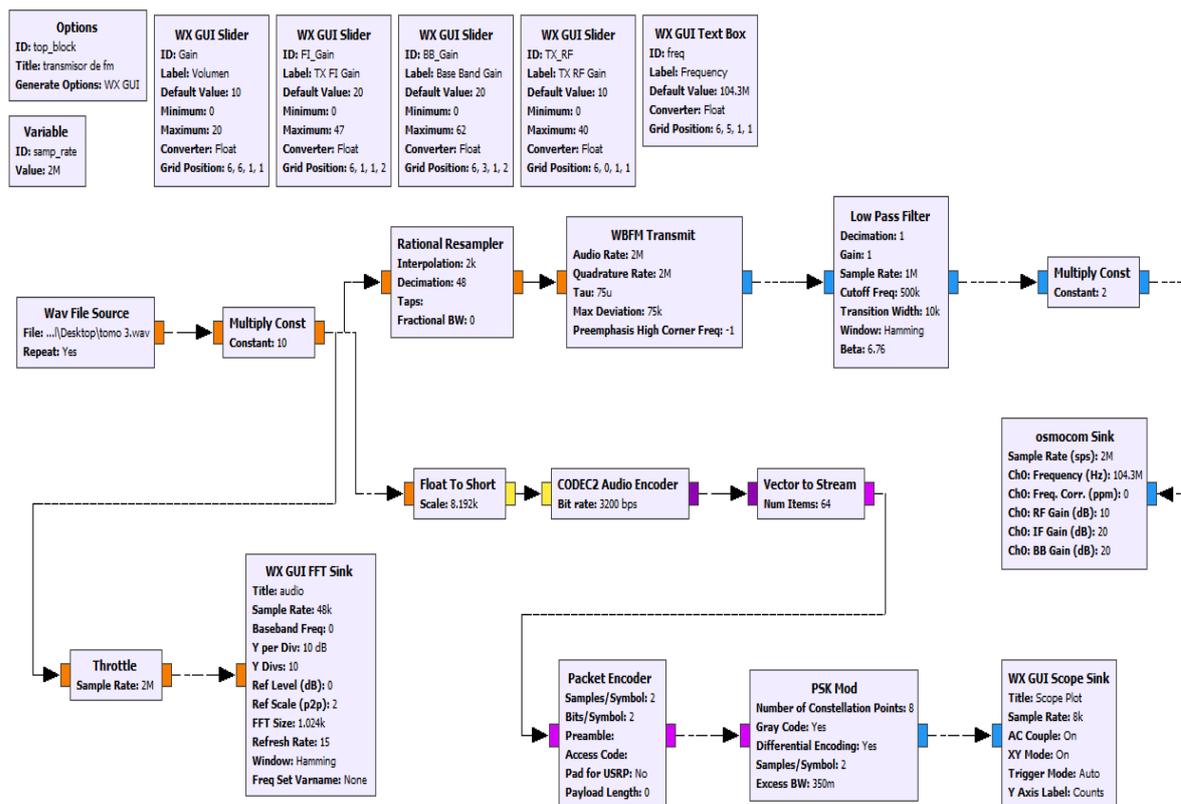
Nota. Elaborador por el autor.

3.6 Desarrollo De La Propuesta

3.6.1 Transmisión De PSK

La **Figura. 38** muestra el diagrama general del transmisor para modulaciones de 4,8,16-PSK, muestra que para realizar estas modulaciones se generan diversos puntos, respecto a cada una de sus constelaciones.

Figura. 38 diagrama de transmisor para modulaciones de 8-PSK

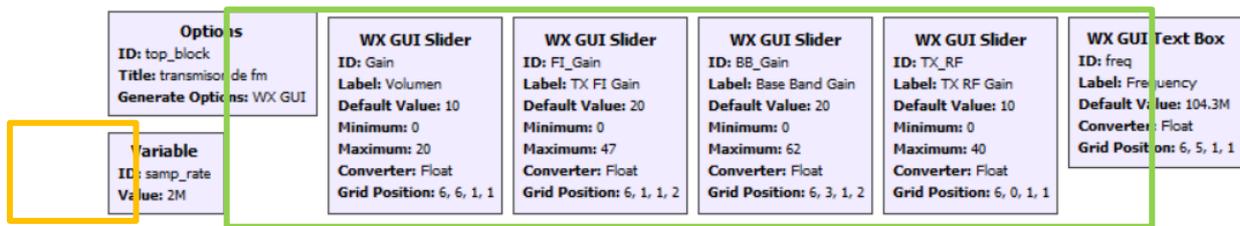


Nota. Elaborador por el autor.

3.6.1.1 Variables y librerías

Se proporciona un entorno de desarrollo amigable al usuario con una biblioteca de bloques de procesamiento, con las que se pueden crear variedad de aplicaciones en GNU Radio, se utilizan varias variables y bibliotecas básicas para facilitar la modificación y el procesamiento de señales, así como se observa en la **Figura. 39**.

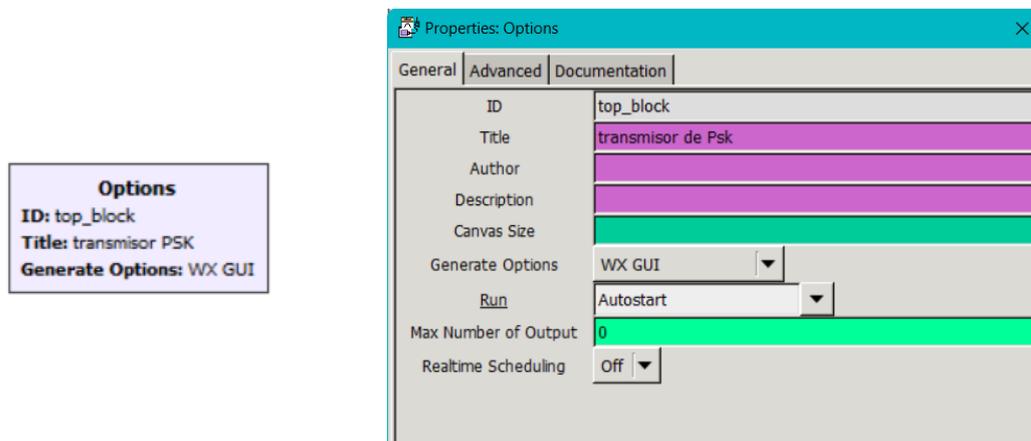
Figura. 39 Variables y librerías de PSK



Nota. A través del gráfico se puede apreciar la variable `sample_rate` que está en el recuadro naranja, en la cual por defecto y se establece en 250K lo que es la tasa de muestreo. Las librerías `Wx Slider` y `Text Box` que están en el recuadro verde esto representa los valores mínimos y máximos que le permite variar al usuario, Elaborador por el autor.

Options: Este módulo que se muestra en la **Figura. 40**, se utiliza para dar información sobre el nombre del proyecto y el nombre de la persona que realiza el esquema. Esto es opcional.

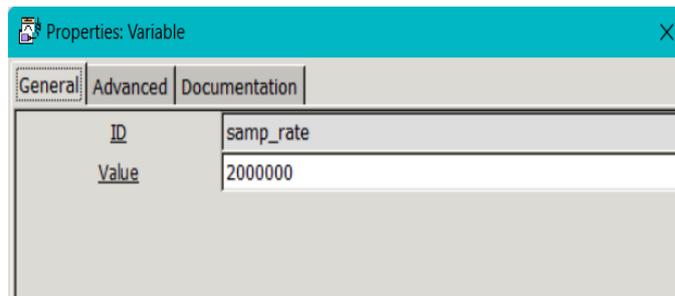
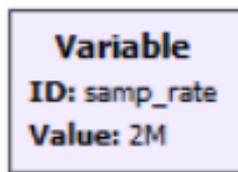
Figura. 40 Bloque Options y Propiedades.



Nota. Elaborador por el autor.

Variable: Como su nombre lo indica son las variables creadas por el usuario, en este caso en la **Figura. 41** se creó el `samp_rate` que es la tasa de muestreo y que se valúa en **2000000** muestras por segundo.

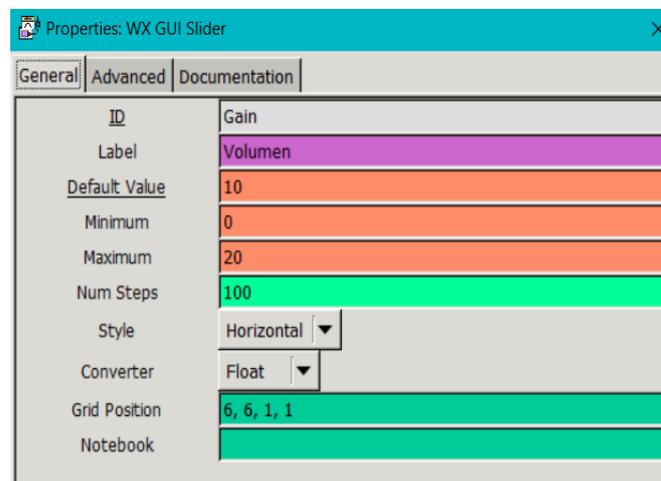
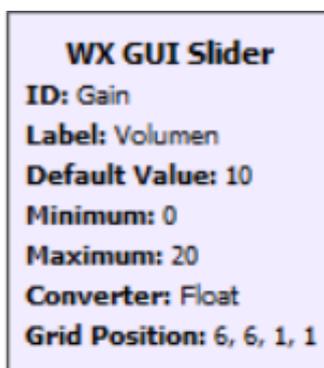
Figura. 41 Bloque Variable y Propiedades.



Nota. Elaborador por el autor.

Wx Gui Slider: El bloque que se observa en la **Figura. 42** nos permite cambiar la ganancia de la señal, presentándose en la interface como el control deslizante debajo de la señal que se está analizando.

Figura. 42 Bloque Wx Gui Slider y Propiedades.



Nota. Elaborador por el autor.

El bloque que se observa en la **Figura. 43** nos permite cambiar la ganancia de la transmisión de la frecuencia intermedia (FI), que evalúa en 20 con un mínimo de 0 y un máximo de 47.

Figura. 43 FI_Gain y Propiedades.

WX GUI Slider
ID: FI_Gain
Label: TX FI Gain
Default Value: 20
Minimum: 0
Maximum: 47
Converter: Float
Grid Position: 6, 1, 1, 2

Property	Value
ID	FI_Gain
Label	TX FI Gain
Default Value	20
Minimum	0
Maximum	47
Num Steps	100
Style	Vertical
Converter	Float
Grid Position	6, 1, 1, 2
Notebook	

Nota. Elaborador por el autor.

En la **Figura. 44** nos permite cambiar la ganancia de la Banda Base (BB), que evalúa en 20 con un mínimo de 0 y un máximo de 62.

Figura. 44 Ganancia de la Banda Base y Propiedades.

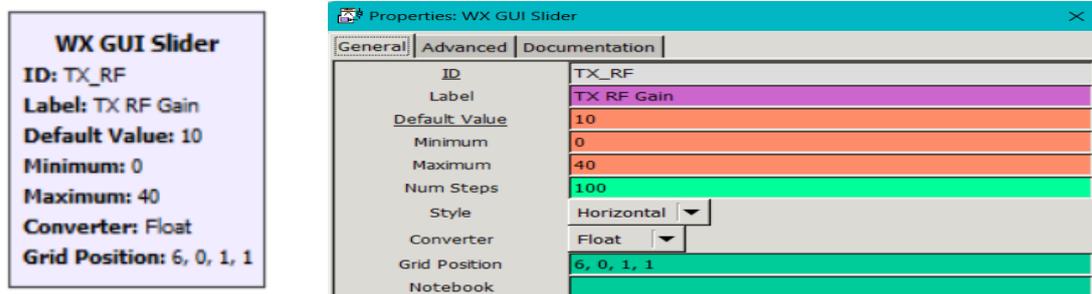
WX GUI Slider
ID: BB_Gain
Label: Base Band Gain
Default Value: 20
Minimum: 0
Maximum: 62
Converter: Float
Grid Position: 6, 3, 1, 2

Property	Value
ID	BB_Gain
Label	Base Band Gain
Default Value	20
Minimum	0
Maximum	62
Num Steps	100
Style	Horizontal
Converter	Float
Grid Position	6, 3, 1, 2
Notebook	

Nota. Elaborador por el autor.

En la **Figura. 45** la ganancia de la transmisión de radio frecuencia (RF) nos permite variar evalúa en 10 con un mínimo de 0 y un máximo de 40.

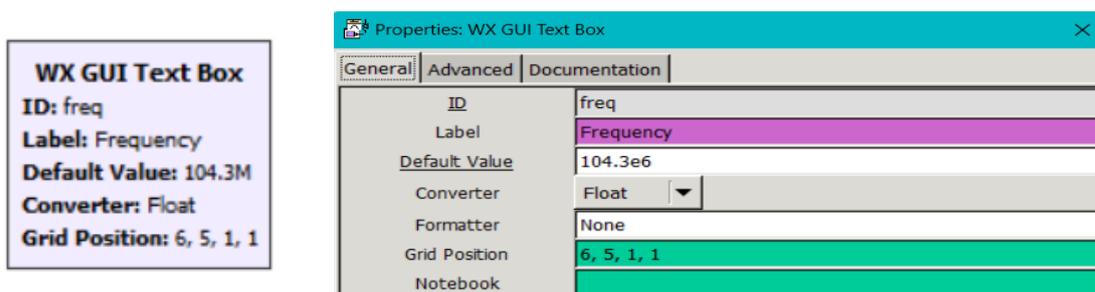
Figura. 45 Ganancia de Radio frecuencia (RF) y Propiedades.



Nota. Elaborador por el autor.

Wx Gui Text Box: Esté bloque permite cambiar el valor de la frecuencia al momento de simular, como se observa en la **Figura. 46** se está trabajando en la frecuencia de 104.3Mhz.

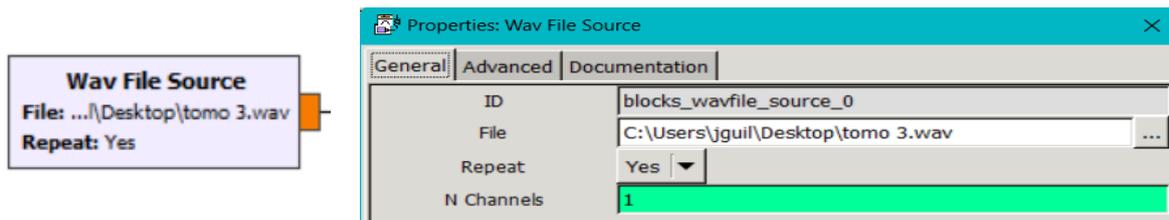
Figura. 46 Bloque Wx Gui Text Box y Propiedades.



Nota. Elaborador por el autor.

Wav File Source: Este bloque analiza archivos de audio .wav y presenta una salida de datos en float, en la **Figura. 47** se observa la ubicación el audio .wav en la cual, va a salir atreves del canal 1.

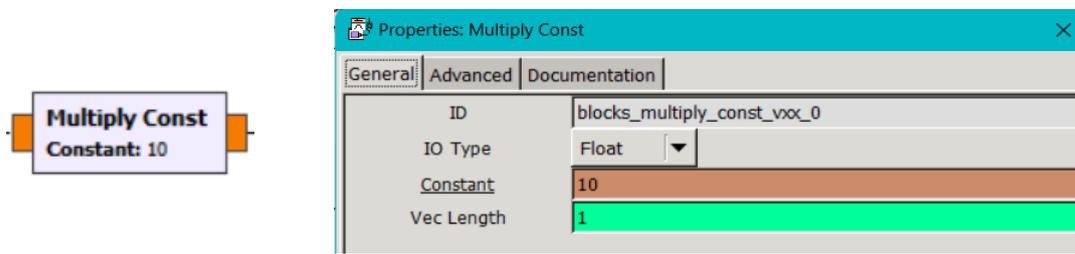
Figura. 47 Bloque Wav File Source y Propiedades.



Nota. Elaborador por el autor.

Multiply Const: Sirve para multiplicar el flujo de entrada (float), como se observa en la **Figura. 48** el flujo se lo multiplica por una contante, que en este caso es 10.

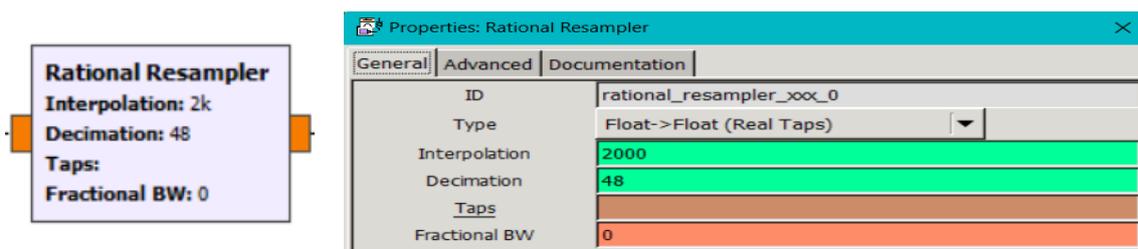
Figura. 48 Bloque Multiply Const y Propiedades.



Nota. Elaborador por el autor.

Rational Resampler: es el que cambia la frecuencia de muestreo, En la **Figura. 49** la configuración del bloque presenta un cambio de la frecuencia de muestreo que va desde 48 a 2000 con el Ancho de banda fraccional en 0.

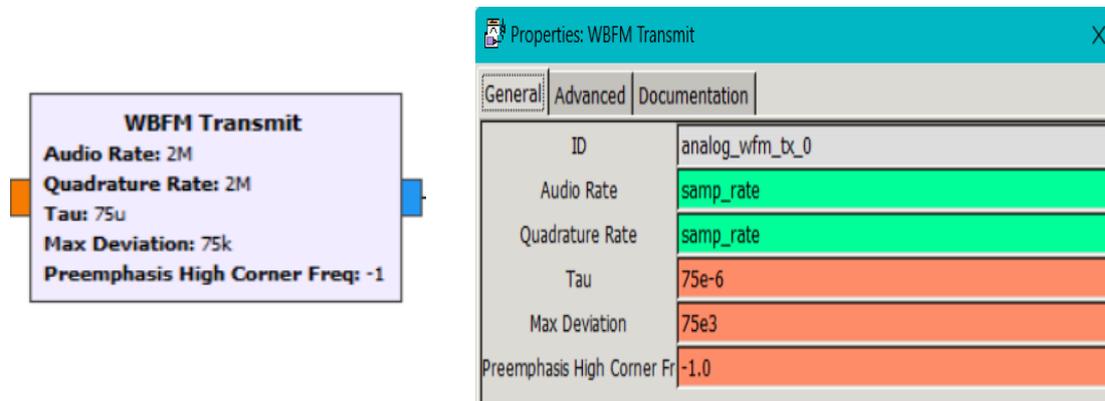
Figura. 49 Bloque Rational Resampler y Propiedades.



Nota. Elaborador por el autor.

WBFM Transmit: El transmisor de banda ancha recepta un valor float de muestras y presenta una salida complex de banda base, como se aprecia en la **Figura. 50** el audio rete y la quadrature rete es el mismo de 2000000, teniendo en cuenta que los demás valores son predeterminados.

Figura. 50 Bloque WBFM Transmit y Propiedades.



Nota. Elaborador por el autor.

Low Pass Filter: Este bloque trabaja como un filtro pasa bajo, como se visualiza en la **Figura. 51** la entrada y salida del bloque es compleja, presentando tasa de decimación del filtro que es 1.

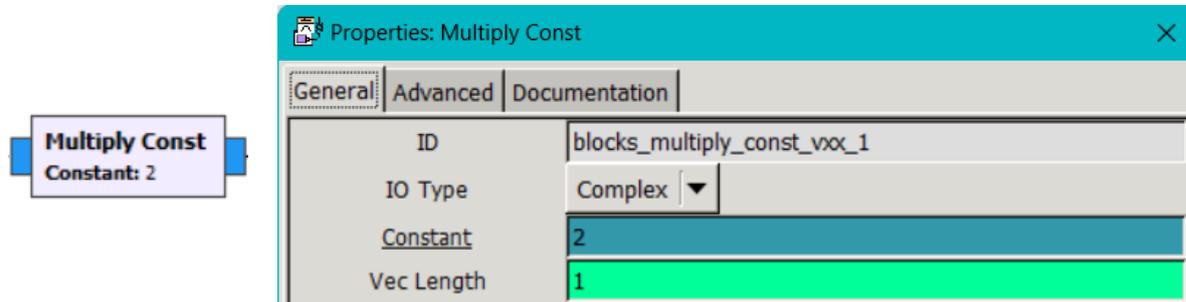
Figura. 51 Bloque Low Pass Filter y Propiedades.



Nota. Elaborador por el autor.

El bloque Multiply Const de la **Figura. 52** presenta un ingreso y salida de datos en complex con una contante de 2.

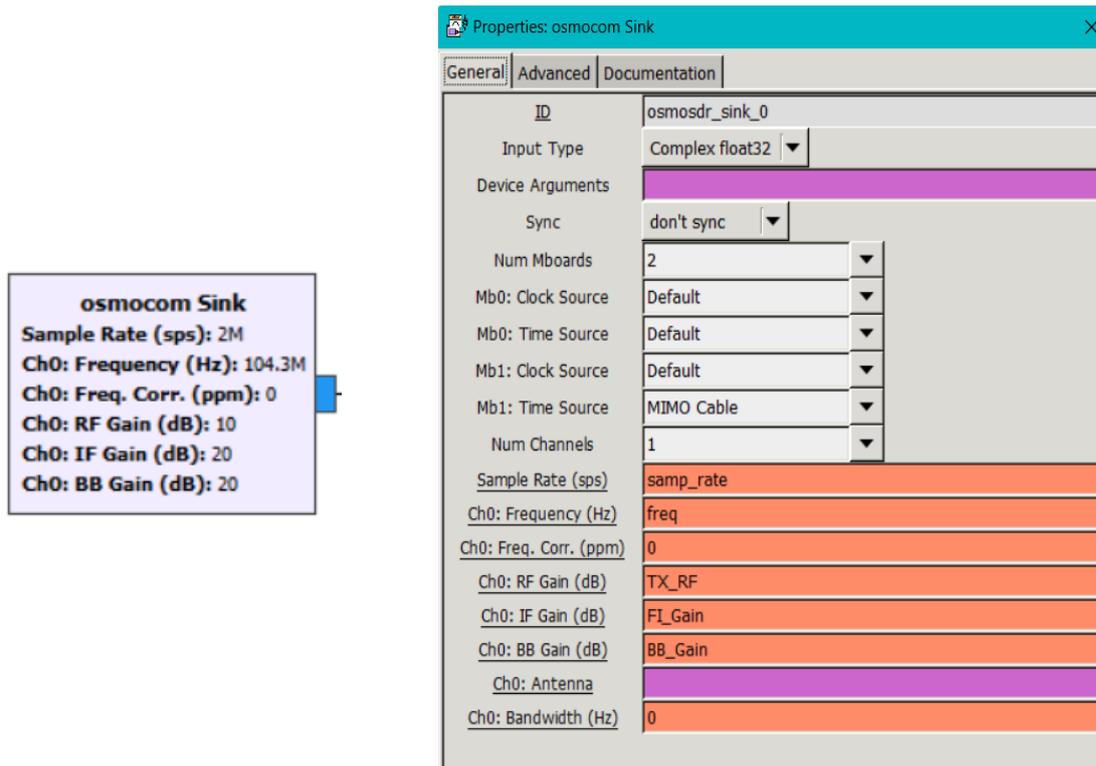
Figura. 52 Bloque Multiply Const con complex y Propiedades.



Nota. Elaborador por el autor.

Osmocom Sink: Es el que transmite la señal ya que, es compatible con el SDR (HackRF One), en la **Figura. 53** se observa las configuraciones de las ganancias.

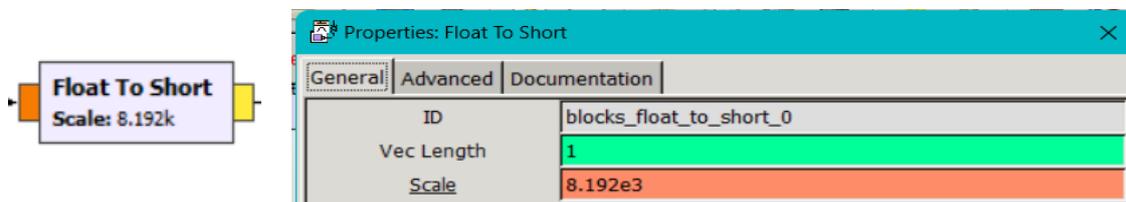
Figura. 53 Bloque Osmocom Sink y Propiedades.



Nota. Elaborador por el autor.

Float To Short: convierte de un float a short, en la **Figura. 54** se observa que esta trabajado con un factor escalar de entrada que es de 8.192khz

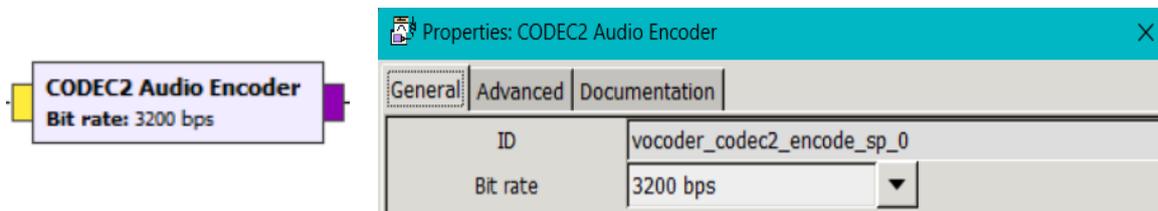
Figura. 54 Bloque Float To Short y Propiedades.



Nota. Elaborador por el autor.

CODEC2 Audio Encoder: Es un codificador de audio, en la **Figura. 55** trabaja con una tasa de datos de 3200 bps.

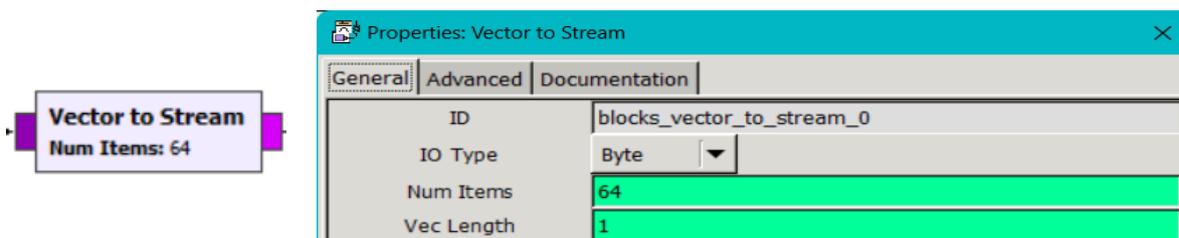
Figura. 55 Bloque CODEC2 Audio Encoder y Propiedades.



Nota. Elaborador por el autor.

Vector to Stream: Convierte vectores de longitud, en la **Figura. 56** se observa el valor del flujo de salida que es de 64 y la longitud del vector de salida que es 1.

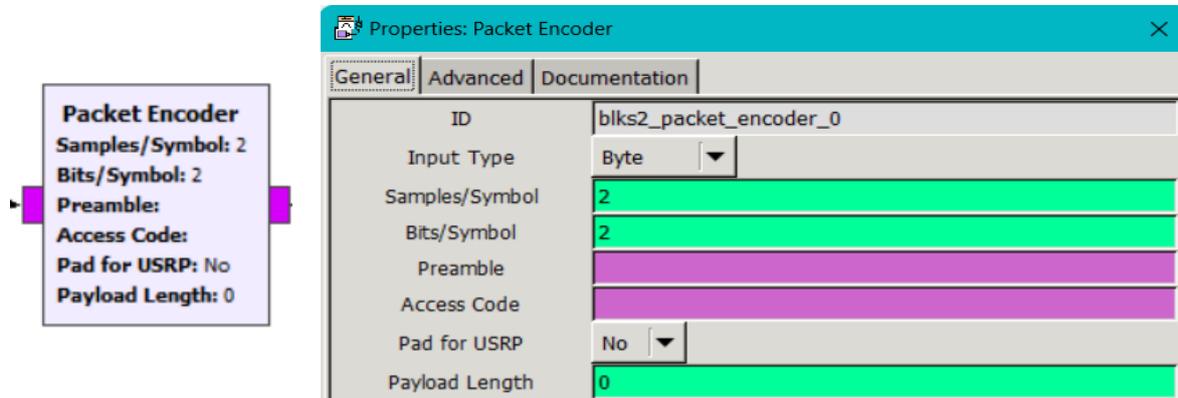
Figura. 56 Bloque Vector to Stream y Propiedades.



Nota. Elaborador por el autor.

Packet Encoder: Sirve para convertir el flujo de datos que ingresan en paquetes, en los parámetros de la **Figura. 57** el símbolo de muestreo y de bits es de 2.

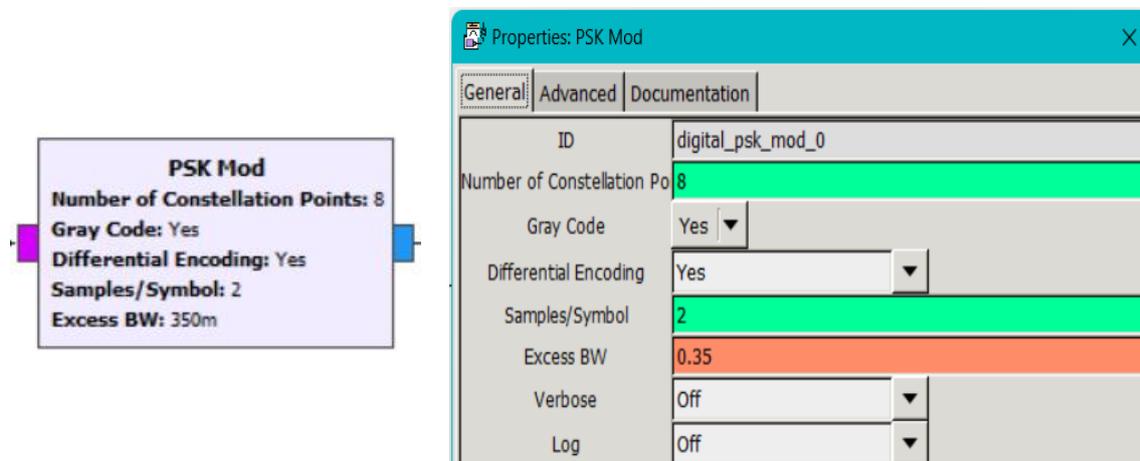
Figura. 57 Bloque Packet Encoder y Propiedades.



Nota. Elaborador por el autor.

PSK Mod: En este bloque trabaja con una señal de ingreso en bytes y de salida en complex, en la **Figura. 58** se puede observar y modificar el número de constelaciones en este caso está 8-PSK y se puede visualizar el símbolo que es 2.

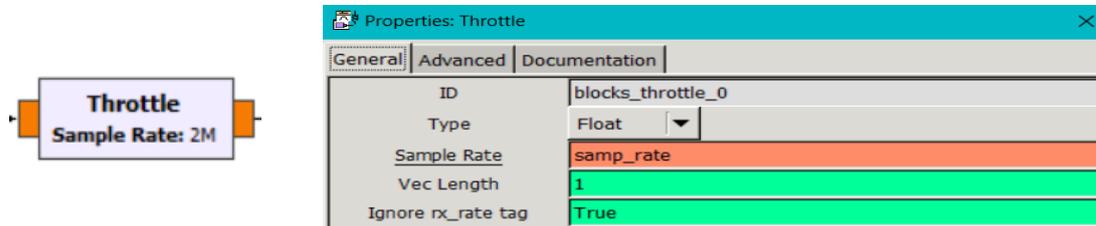
Figura. 58 Bloque PSK Mod y Propiedades.



Nota. Elaborador por el autor.

Throttle: Se lo utiliza para cambiar el tipo de variable, en la **Figura. 59** se visualiza que se está trabajando con datos tipo float y que el valor de la frecuencia de muestreo es de 2000000 Hz.

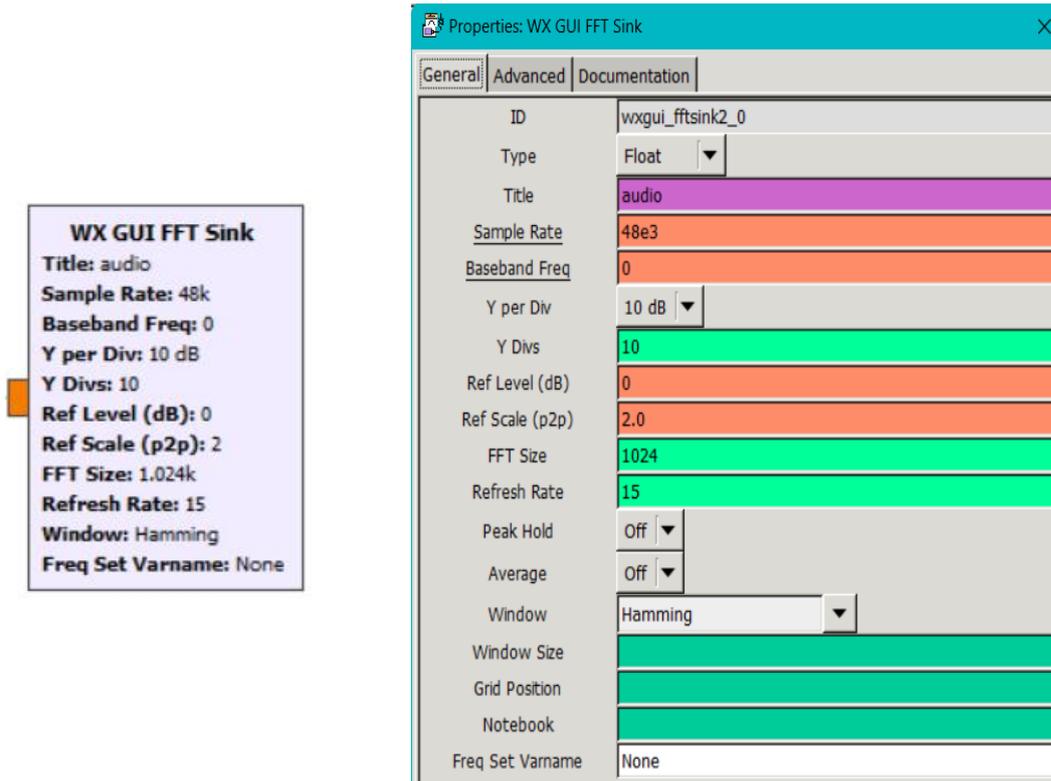
Figura. 59 Bloque Throttle y Propiedades.



Nota. Elaborador por el autor.

WX GUI FFT Sink :Este bloque genera una gráfica de la señal de entrada, en la **Figura. 60** se observa las configuraciones de los parámetros del bloque.

Figura. 60 Bloque WX GUI FFT Sink y Propiedades.



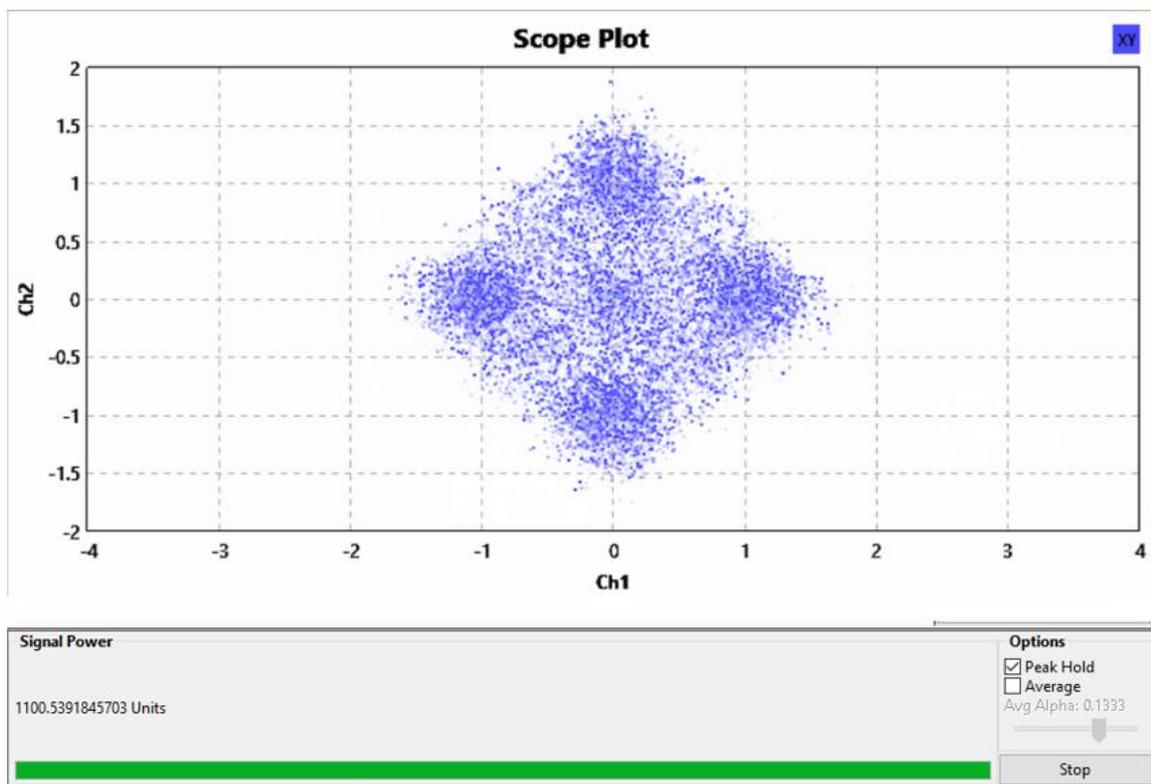
Nota. Elaborador por el autor.

Un diagrama de constelación es una representación gráfica de una señal proyectada en un espacio vectorial ortogonal, cuyas dimensiones están determinadas por el tipo particular de modulación. Las constelaciones bidimensionales son las más comunes y pueden representar los diferentes tipos de modulación.

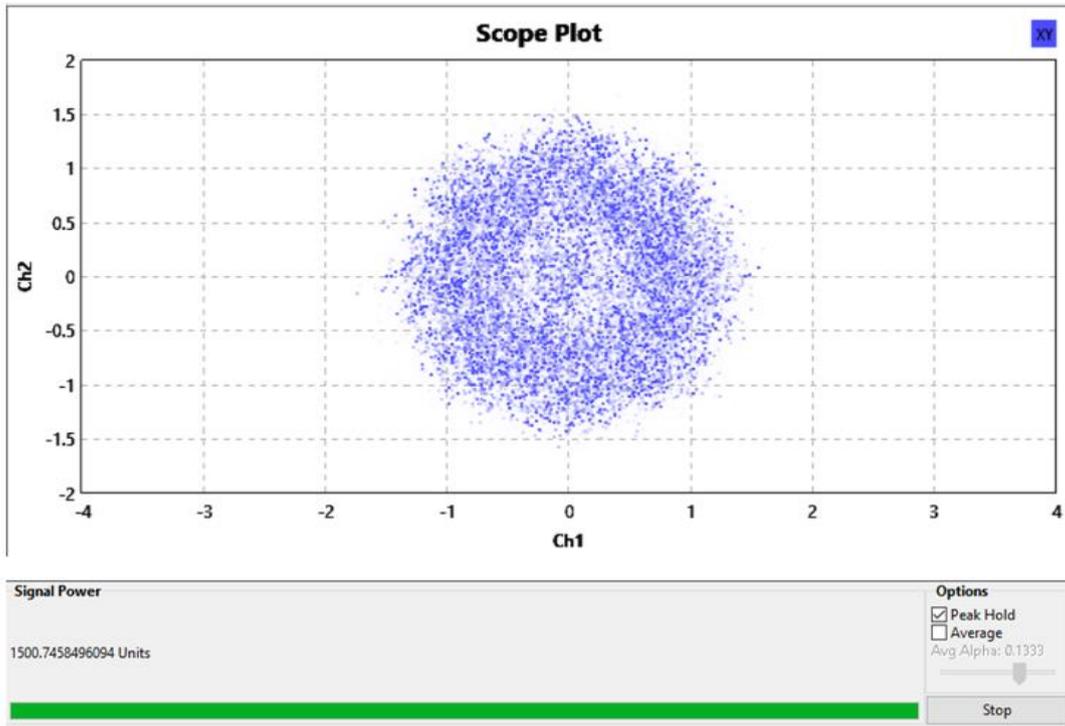
Este proyecto proporciona los diagramas de constelación para la señal 4,8,16-PSK, lo que permite analizar la calidad de la señal y detectar posibles problemas en la transmisión.

Como se puede ver en la **Figura. 61**, las características del gráfico de constelación están conformado con los siguientes valores 10 dBm de TX RF Gain, 20 dB de TX FI Gain, y 20 dB de Base Band Gain están proporcionando una señal de audio amplificada en la banda de frecuencia de 104.3 MHz.

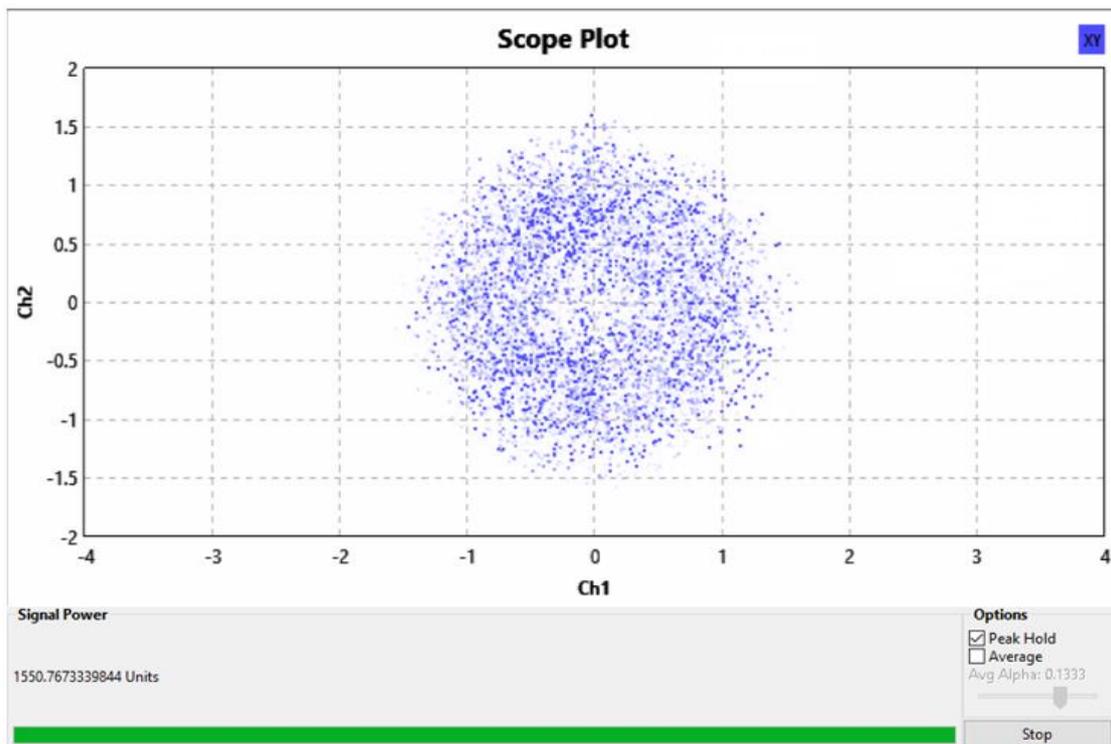
Figura. 61 Resultado de Constelaciones a) 4-PSK. b) 8- PSK. c) 16- PSK.



a)



b)



c)

Nota. Elaborador por el autor.

Cálculos de la SNR de las modulaciones 4,8,16 PSK

Para realizar el cálculo de la SNR primero se halla la potencia del ruido utilizando la Ecuación

4 :

$$P_{Ruido} = P_{Rx} - P_{Tx} \quad \text{Ecuación 4}$$

Donde:

P_{Tx} = Es la potencia de la señal transmitida (sin ruido).

P_{Rx} = Potencia de la señal recibida (con ruido).

Datos de 4 PSK:

$$P_{Tx} = 792u$$

$$P_{Rx} = 1100u$$

$$P_{Ruido} = 1100u - 792u = \mathbf{308u}$$

Una vez encontrada la potencia del ruido y teniendo la potencia de la señal transmitida se procede a utilizar la Ecuación 5 para obtener la SNR de la transmisión 4, 8 y 16 PSK:

$$SNR_{QPSK}(dB) = 10 \log_{10} \left(\frac{P_{Tx}}{P_{Ruido}} \right) \quad \text{Ecuación 5}$$

$$SNR_{4PSK}(dB) = 10 \log_{10} \left(\frac{792u}{308u} \right) = \mathbf{4.10 dB}$$

Datos de 8 PSK :

$$P_{Tx} = 792u$$

$$P_{Rx} = 1500u$$

$$P_{Ruido} = 1500u - 792u = \mathbf{708u}$$

$$SNR_{8PSK}(dB) = 10\text{Log}_{10}\left(\frac{792u}{708u}\right) = \mathbf{0.49 dB}$$

Datos de 16 PSK:

$$P_{Tx} = 792u$$

$$P_{Rx} = 1550u$$

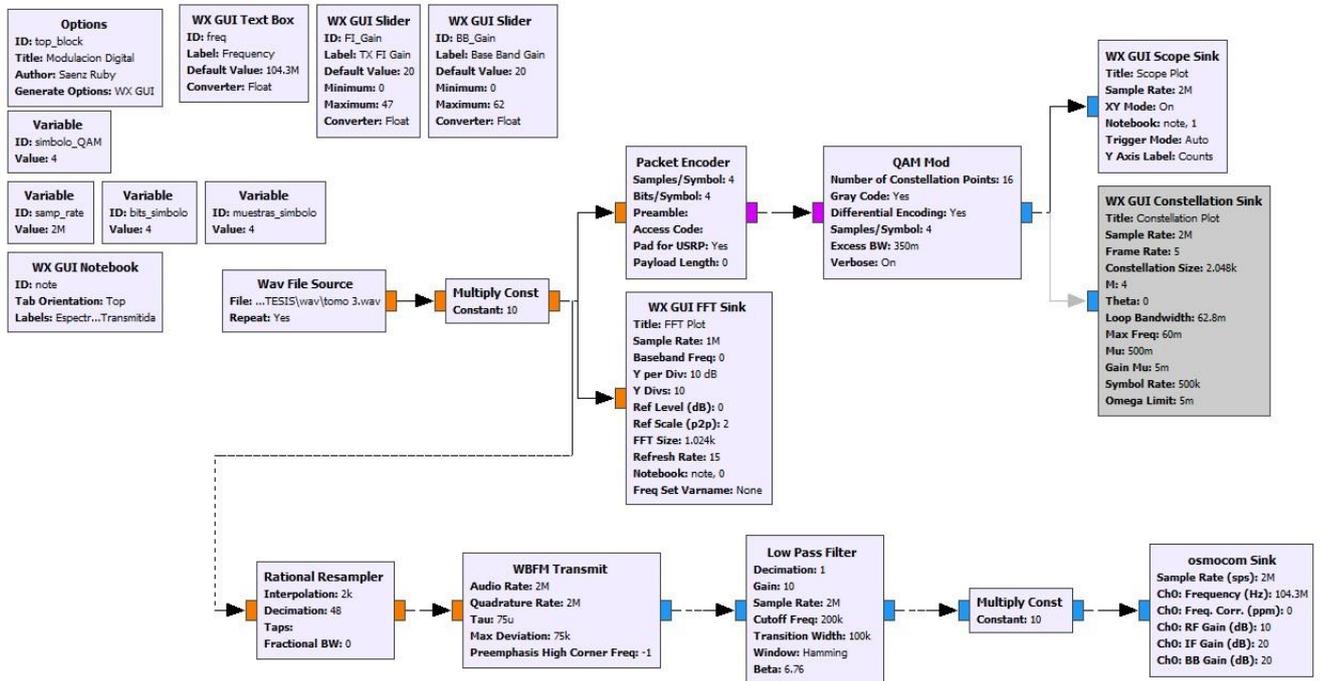
$$P_{Ruido} = 1550u - 792u = \mathbf{758u}$$

$$SNR_{16PSK}(dB) = 10\text{Log}_{10}\left(\frac{792u}{758u}\right) = \mathbf{0.2 dB}$$

3.6.2 Transmisión De QAM

En la **Figura. 62** se presenta el diagrama del transmisor para las modulaciones 16,64-QAM que están trabajando en la frecuencia de 104.3Mhz.

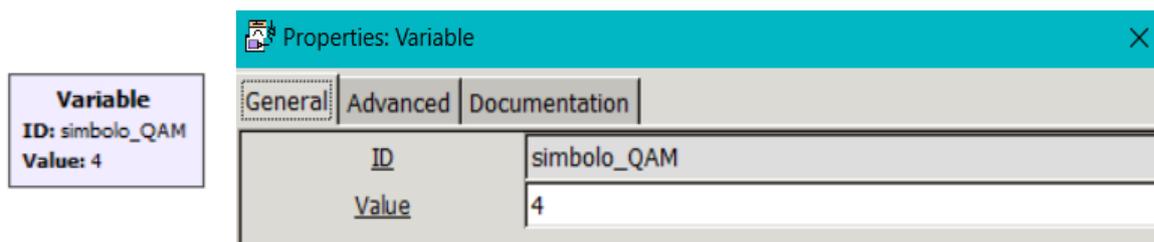
Figura. 62 Diagrama de transmisión de las modulaciones 16 QAM.



Nota. Elaborador por el autor.

En la variable Símbolo_QAM de la **Figura. 63** se observa que está evaluando en 4 , este valor varia acorde a la modulación que se trabaje.

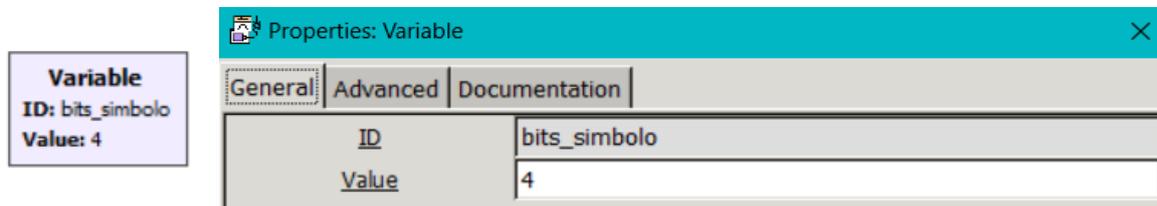
Figura. 63 Variable Símbolo_QAM y Propiedades.



Nota. Elaborador por el autor.

En la **Figura. 64** la variable Símbolo de Bit se está evaluando en 4 basado en la **Tabla 1**.

Figura. 64 Variable Bits/Símbolo y Propiedades.



Nota. Elaborador por el autor.

La **Figura. 65** muestra la variable Símbolo de muestras que trabaja en 4 bits.

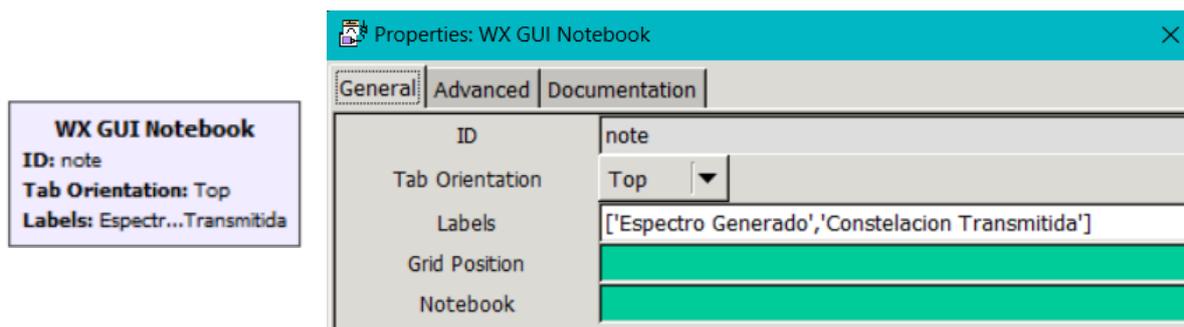
Figura. 65 Variable Muestras/Símbolo y Propiedades.



Nota. Elaborador por el autor.

WX GUI Notebook: Este bloque puede unificar varias ventanas en una sola, en la **Figura. 66** se observa que va a agrupar las ventanas espectro y constelación.

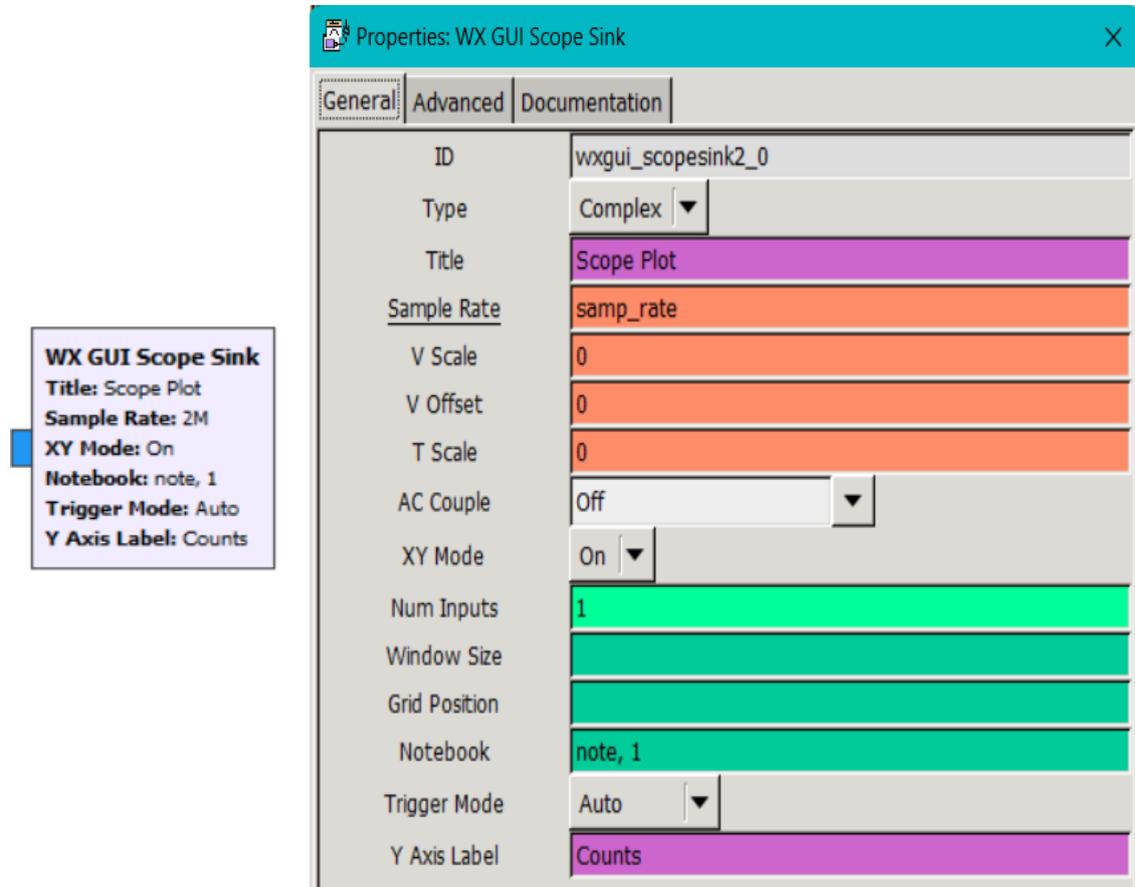
Figura. 66 Bloque WX GUI Notebook y Propiedades.



Nota. Elaborador por el autor.

WX GUI Scope Sink: Es un osciloscopio digital, como se visualiza en la **Figura. 67** la configuración de los parámetros del bloque.

Figura. 67 Bloque WX GUI Scope Sink y Propiedades.



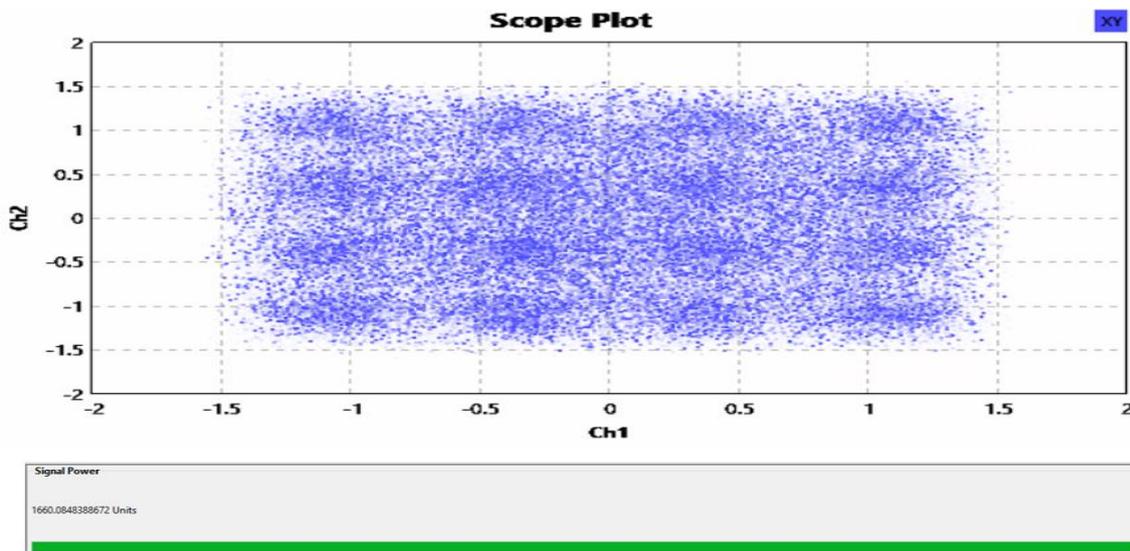
Nota. Elaborador por el autor.

Los resultados que se observan en la **Figura. 68**, proporcionan los diagramas de constelación de la modulación QAM, que se utilizan para representar los bits en una frecuencia de 104.3 Mhz. Una señal de 16-QAM se representa por un total de 16 zonas diferentes y cada punto representa una combinación específica de 4 bits cómo se visualiza en la **Figura. 68 a)**, al haber menos puntos en la constelación estos están más separados, por lo tanto, son más fáciles de distinguir entre sí.

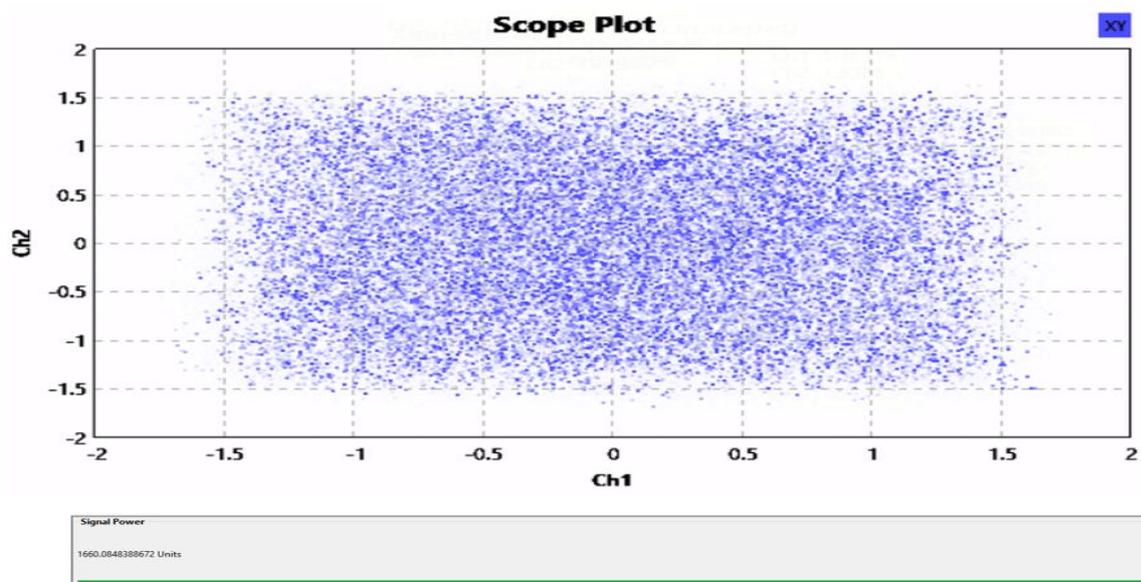
Por otro lado, 64-QAM se representa por un diagrama de 64 zonas ya que, estos puntos están más cercanos entre sí, lo que significa que la distancia entre cada punto

es menor, como se aprecia en la **Figura. 68 b)**, Esta cercanía hace que los puntos sean más difíciles de distinguir, especialmente en entornos donde hay ruido o interferencias, lo que puede llevar a una mayor probabilidad de errores en la interpretación de los símbolos.

Figura. 68 Diagrama de Constelaciones resultado a) 16-QAM. b) 64-QAM.



a)



b)

Nota. Elaborador por el autor.

Cálculos de la SNR de las modulaciones 16 y 64 QAM

Para realizar el cálculo de la señal relación- ruido (SNR) primero se halla la potencia del ruido utilizando la Ecuación 4 y una vez encontrada la potencia del ruido y teniendo la potencia de la señal transmitida se utiliza la Ecuación 5 para obtener la SNR de la transmisión de 16 y 64 QAM:

Datos de 16 QAM:

$$P_{Tx} = 792u$$

$$P_{Rx} = 1168u$$

$$P_{Ruido} = 1168u - 792u = \mathbf{376u}$$

$$SNR_{16QAM}(dB) = 10\text{Log}_{10}\left(\frac{792u}{376u}\right) = \mathbf{3.24 dB}$$

Datos de 64 QAM :

$$P_{Tx} = 792u$$

$$P_{Rx} = 1660u$$

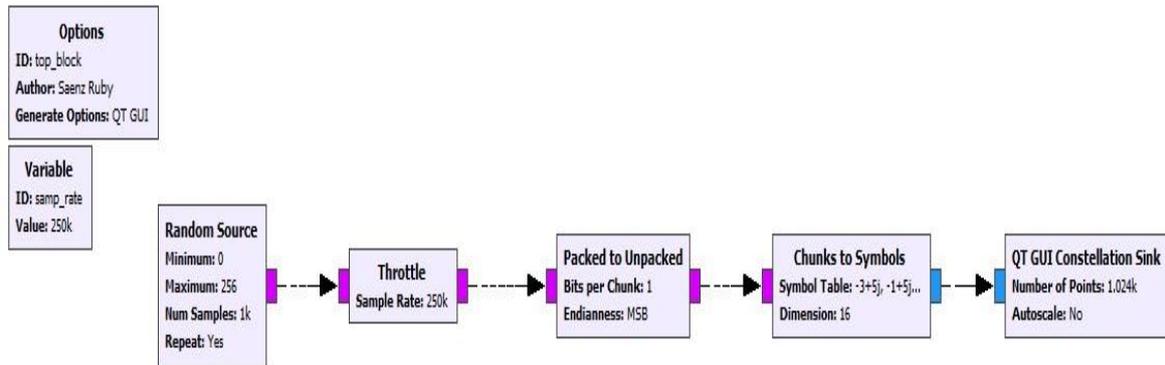
$$P_{Ruido} = 1660u - 792u = \mathbf{868u}$$

$$SNR_{64QAM}(dB) = 10\text{Log}_{10}\left(\frac{792u}{868u}\right) = \mathbf{-0.40 dB}$$

3.6.3 Modulación de 32-QAM

En la **Figura. 69** se presenta el diagrama de la modulación 32-QAM, donde se puede observar su configuración de los bloques.

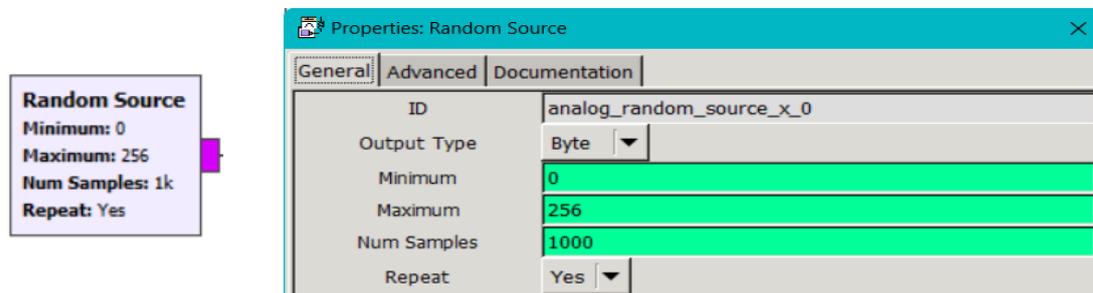
Figura. 69 Diagrama de la modulación de 32-QAM.



Nota. Elaborador por el autor.

Random Source: Es el generador de números aleatorios como se observa en la **Figura. 70** se realiza entre 0 y 256 con una muestra de 1000 para el cálculo del (SNR).

Figura. 70 Bloque Random Source y Propiedades.

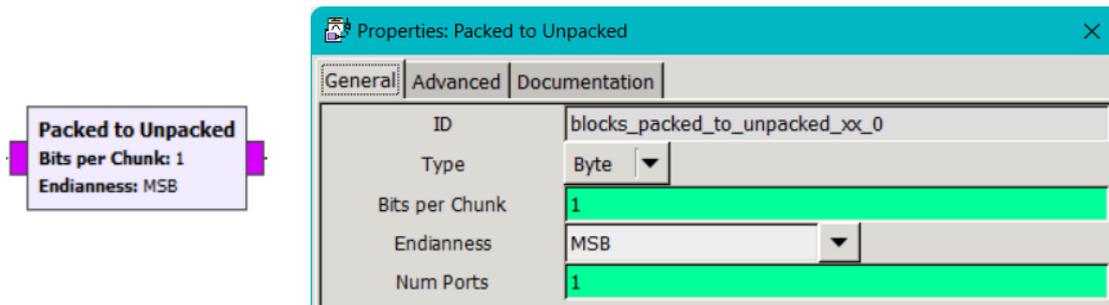


Nota. Elaborador por el autor.

Packed to Unpacked: Convierte los valores que fueron generados en el bloque de la **Figura. 70**, los agrupa en fragmentos de símbolos de 1 byte, utilizando el bit más significativo (MSB) y con 1 puerto de entrada y salida como se observa en la **Figura.**

71.

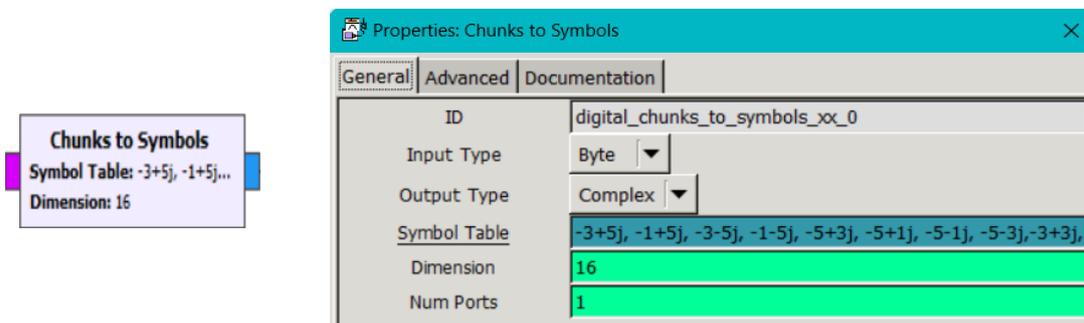
Figura. 71 Bloque Packed to Unpacked y Propiedades.



Nota. Elaborador por el autor.

Chunks to Symbols: Este bloque convierte los símbolos de entrada de 1 byte a símbolos de salida complex, utilizando la tabla de símbolos que se muestra en la **Figura. 72** con una dimensión de 16 en la gráfica.

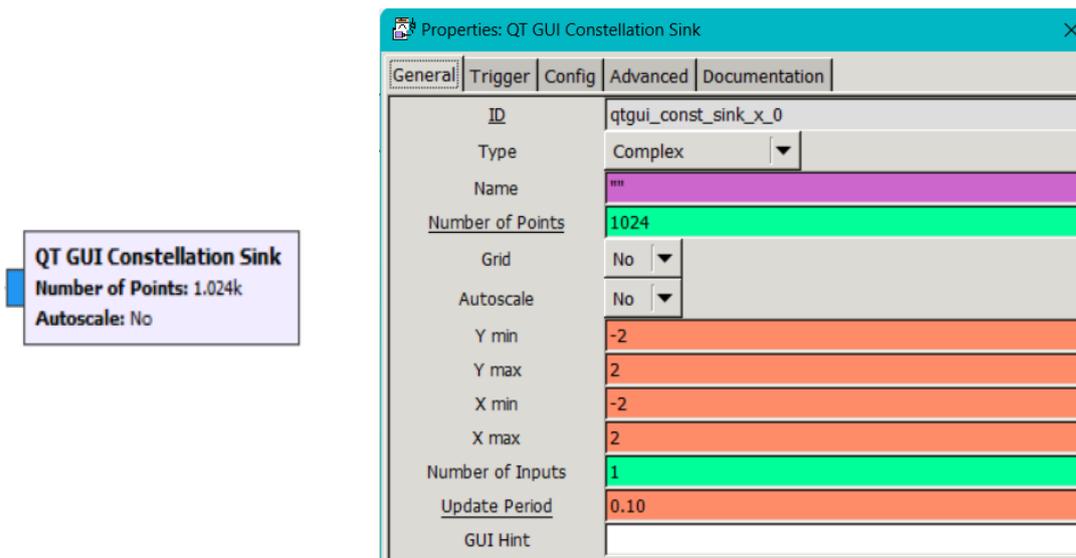
Figura. 72 Bloque Chunks to Symbols y Propiedades.



Nota. Elaborador por el autor.

QT GUI Constellation Sink: En la configuración de las propiedades de **Figura. 73** como resultado presenta en una ventana gráfica los símbolos completos en forma de constelación.

Figura. 73 Bloque QT GUI Constellation Sink y Propiedades.



Nota. Elaborador por el autor.

En el caso de la **Figura. 75**, la modulación de 32-QAM ideal, se utiliza 32 símbolos y cada símbolo representa una combinación de 5 bits como se observa en la Ecuación 6.

$$\text{Número de símbolos: } (2^5) = 32$$

Ecuación 6

Una vez teniendo en claro la modulación de 32-QAM facilita la interpretación de las señal. A continuación, se muestra la **Tabla 6**, de los valores de 32-QAM ideal :

Tabla 6 Modulación 32-QAM

Bits					Símbolo	I	Q
0	0	0	0	0	S1	-3	+5
0	0	0	0	1	S2	-1	+5
0	0	0	1	0	S3	-1	-5
0	0	0	1	1	S4	-3	-5

0	0	1	0	0	S5	-5	+3
0	0	1	0	1	S6	-5	+1
0	0	1	1	0	S7	-5	-1
0	0	1	1	1	S8	-5	-3
0	1	0	0	0	S9	-3	+3
0	1	0	0	1	S10	-3	+1
0	1	0	1	0	S11	-3	-1
0	1	0	1	1	S12	-3	-3
0	1	1	0	0	S13	-1	+3
0	1	1	0	1	S14	-1	+1
0	1	1	1	0	S15	-1	-1
0	1	1	1	1	S16	-1	-3
1	0	0	0	0	S17	+1	+3
1	0	0	0	1	S18	+1	+1
1	0	0	1	0	S19	+1	-1
1	0	0	1	1	S20	+1	-3
1	0	1	0	0	S21	+3	+3
1	0	1	0	1	S22	+3	+1
1	0	1	1	0	S23	+3	-1
1	0	1	1	1	S24	+3	-3

1	1	0	0	0	S25	+5	+3
1	1	0	0	1	S26	+5	+1
1	1	0	1	0	S27	+5	-1
1	1	0	1	1	S28	+5	-3
1	1	1	0	0	S29	+3	+5
1	1	1	0	1	S30	+1	+5
1	1	1	1	0	S31	+1	-5
1	1	1	1	1	S32	+3	-5

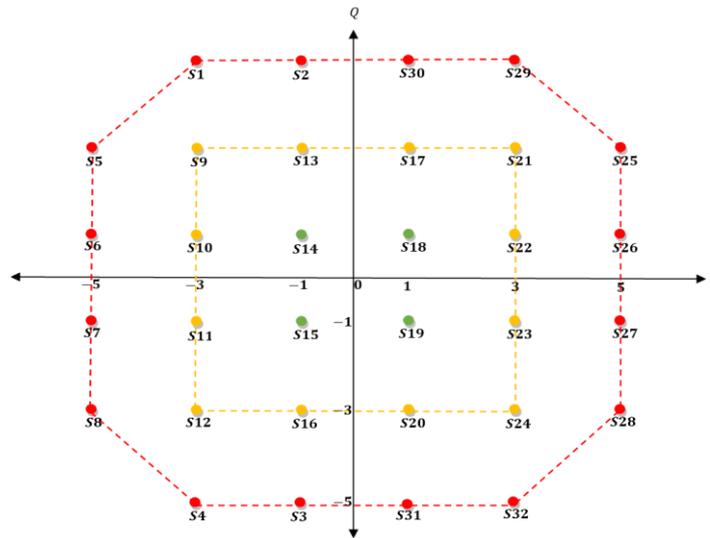
La **Figura. 74** representa los valores de I y Q en un plano cartesiano :

Fase (I): Se refiere a la componente horizontal de la señal. En la constelación, cada punto del eje horizontal (I) representa un determinado valor de la señal moduladora.

Cuadratura (Q): Representa la componente vertical de la señal. En el diagrama de constelación, cada punto en el eje vertical (Q) representa un valor de la señal moduladora.

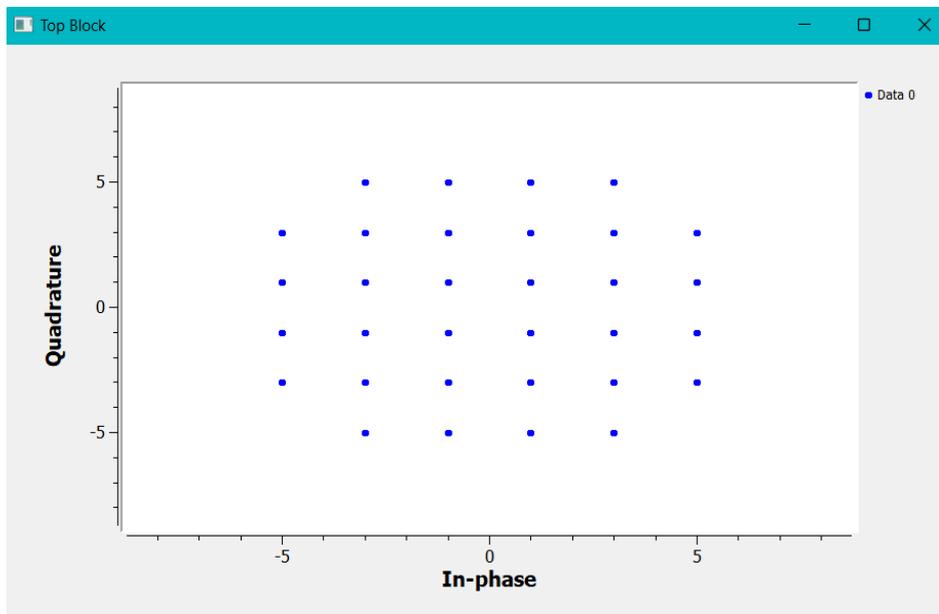
La combinación de valores I y Q en la constelación determina los símbolos de modulación que transmiten la información digital a través de la señal portadora. Una constelación en estos casos es una representación gráfica de símbolos de modulación en el espacio complejo I y Q, donde cada punto o símbolo representa una combinación específica de valores I y Q, como se visualiza en la **Figura. 75** generando una modulación de 32-QAM ideal.

Figura. 74 Diagrama De Constelación con posición en I y Q.



Nota. Elaborador por el autor.

Figura. 75 Diagrama De Constelación, Resultado De 32-QAM Ideal.

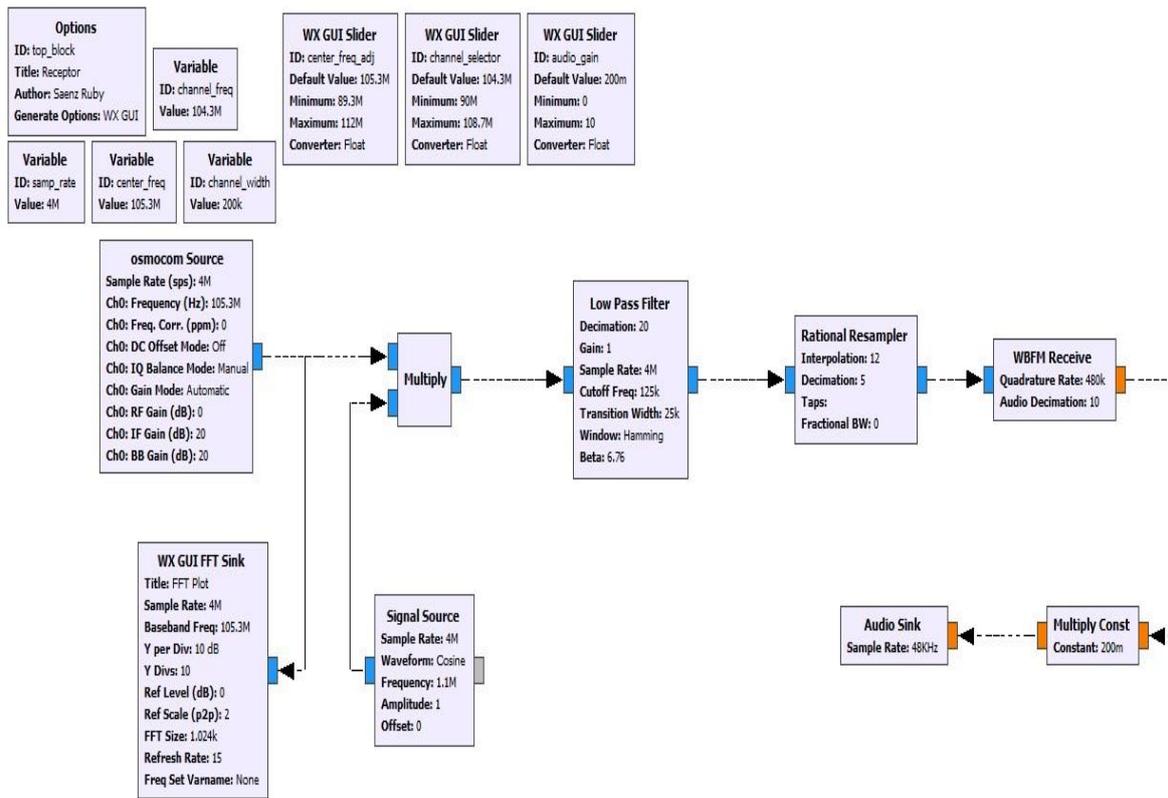


Nota. Elaborador por el autor.

3.6.4 Recepción

El diagrama de la **Figura. 76** es el receptor de las modulaciones de PSK y QAM, sirve para verifica su correcta transmisión de las modulaciones en la frecuencia de 104.3 Mhz.

Figura. 76 Receptor.



Nota. Elaborador por el autor.

Variables: Este bloque viene por defecto el sample_rate, pero si se puede modificar a otro tipo de variables como se muestran a continuación:

- **Center_freq:** trabaja con una frecuencia central de 105.3 Mhz
- **Channel_freq:** la frecuencia del canal es de 104.3 Mhz
- **Channel_width:** El ancho de banda es de 200 KHz

WX GUI Slinder: Este bloque permite variar los valores de un mínimo a un máximo.

- **Center_freq_adj:** Ajusta de la frecuencia central de 105.3 Mhz entre 89.3 Mhz a 112 Mhz
- **Channel_selector:** Permite seleccionar las frecuencias que van desde 90 Mhz a 108.7 Mhz evaluando en 104.3 Mhz.

- **Audio_gain:** La ganancia del audio va desde 0 a 10 y se centra en 200 mhz

Osmocom Source: El bloque que se visualiza en la **Figura. 77** es el que permite recibir la señal mediante el SDR (HackRF One).

Figura. 77 Bloque Osmocom Source y Propiedades.

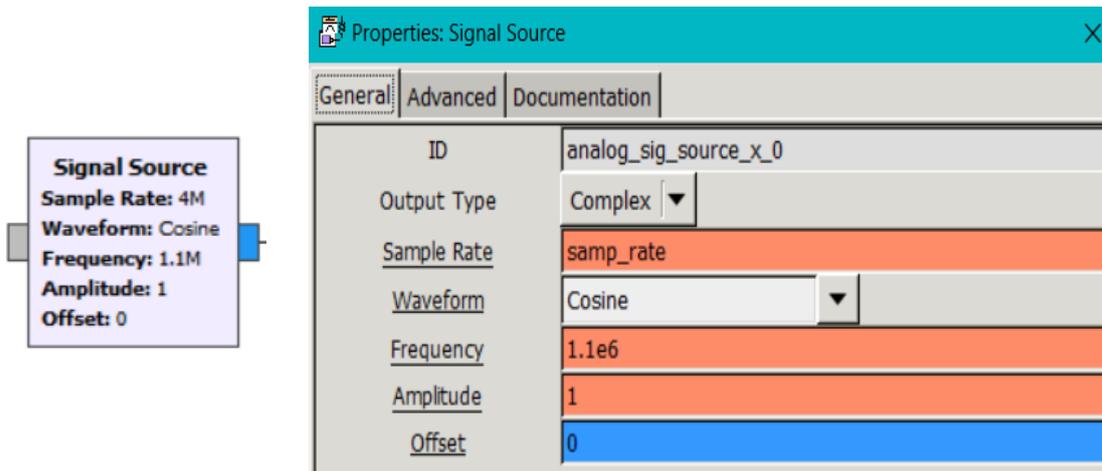
osmocom Source
 Sample Rate (sps): 4M
 Ch0: Frequency (Hz): 105.3M
 Ch0: Freq. Corr. (ppm): 0
 Ch0: DC Offset Mode: Off
 Ch0: IQ Balance Mode: Manual
 Ch0: Gain Mode: Automatic
 Ch0: RF Gain (dB): 0
 Ch0: IF Gain (dB): 20
 Ch0: BB Gain (dB): 20

Properties: osmocom Source	
ID	osmosdr_source_0
Output Type	Complex float32
Device Arguments	
Sync	don't sync
Num Mboards	1
Mb0: Clock Source	Default
Mb0: Time Source	Default
Num Channels	1
Sample Rate (sps)	samp_rate
Ch0: Frequency (Hz)	center_freq
Ch0: Freq. Corr. (ppm)	0
Ch0: DC Offset Mode	Off
Ch0: IQ Balance Mode	Manual
Ch0: Gain Mode	Automatic
Ch0: RF Gain (dB)	0
Ch0: IF Gain (dB)	20
Ch0: BB Gain (dB)	20
Ch0: Antenna	
Ch0: Bandwidth (Hz)	0

Nota. Elaborador por el autor.

Signal Source: En la **Figura. 78** el bloque nos permite escoger el tipo de onda con la que se va dándole una amplitud de 1 y una frecuencia de 1.1 Mhz.

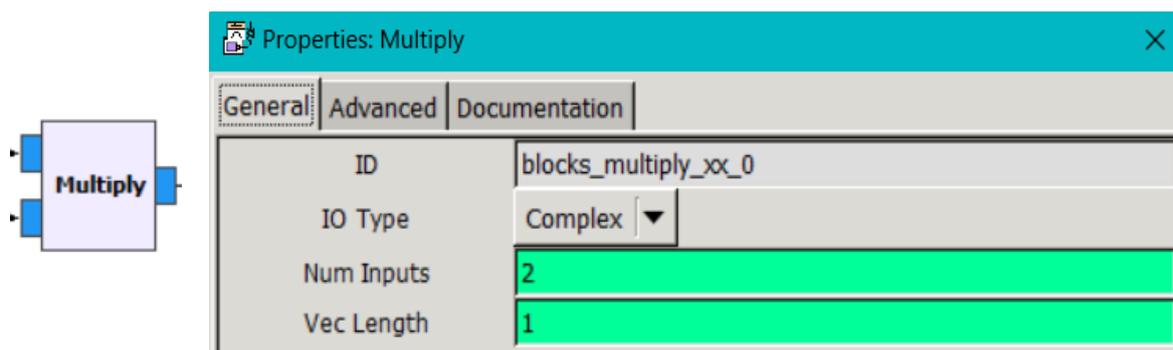
Figura. 78 Bloque Signal Source y Propiedades.



Nota. Elaborador por el autor.

Multiply: Como se observa en la **Figura. 79** recibe el ingreso de dos señales, la señal recibida y la multiplica por una señal generada, en la cual presenta una sola señal de salida.

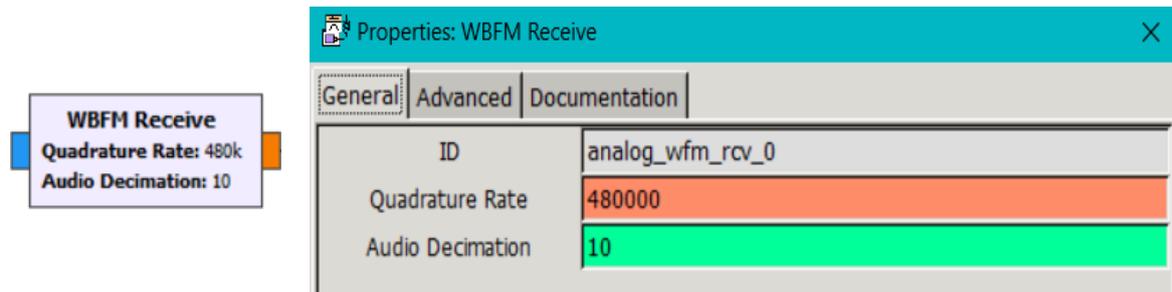
Figura. 79 Bloque Multiply y Propiedades.



Nota. Elaborador por el autor.

WBFM Receiver: Este bloque demodula una señal FM con tipo de dato complex y lo convierte a float, en la **Figura. 80** se aprecia que la tasa de cuadratura con la que trabaja es de 480 Khz.

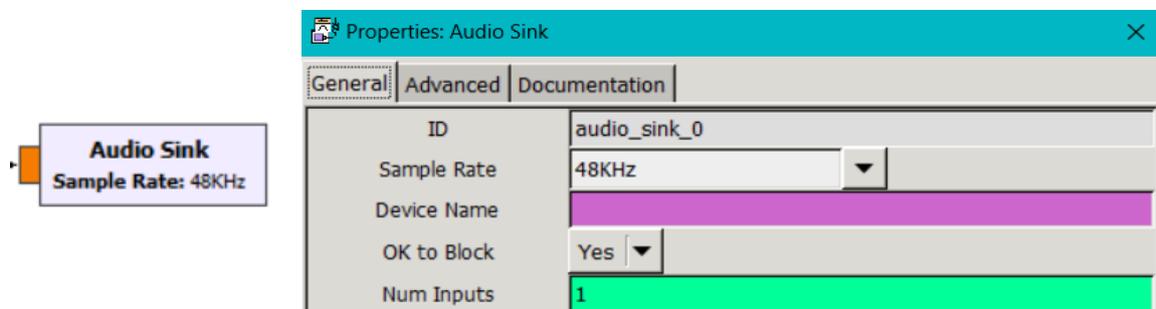
Figura. 80 Bloque WBFM Receiver y Propiedades.



Nota. Elaborador por el autor.

Audio Sink: Este bloque permite interpretar la señal que genero el bloque signal source, cambiándolo a un lenguaje más entendible para el usuario, en la **Figura. 81** se observa que trabaja con una entrada float y con un frecuencia de muestreo de 48 Khz.

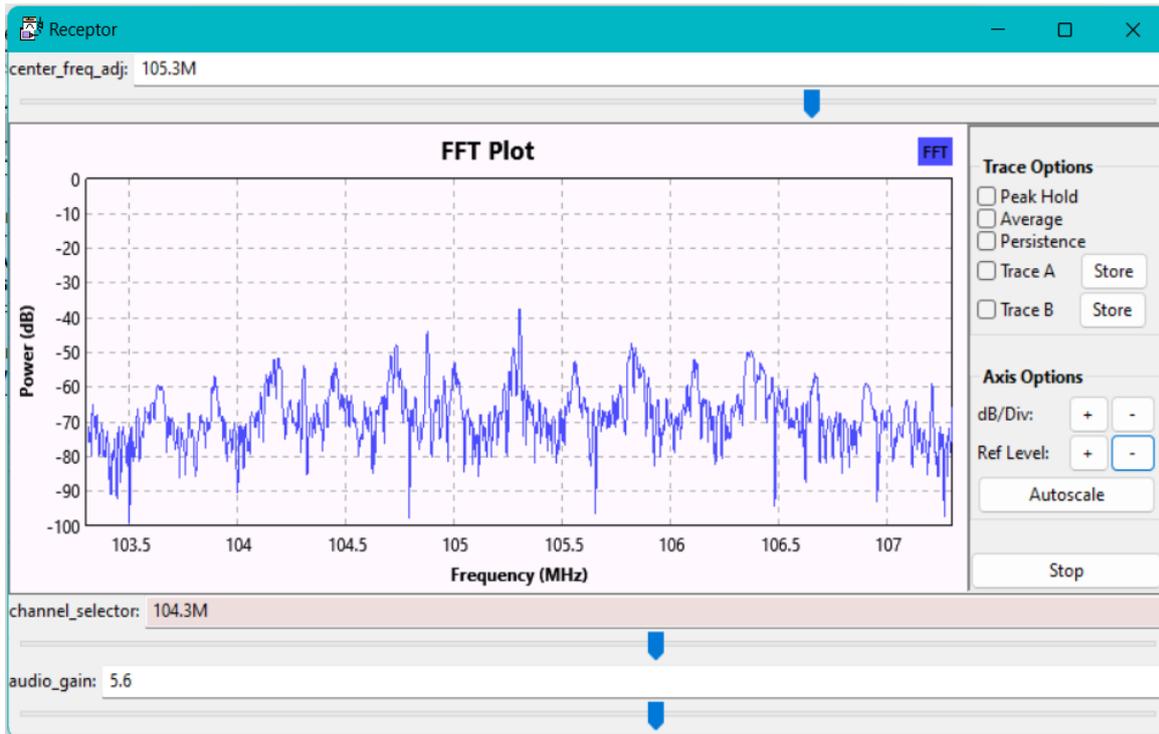
Figura. 81 Bloque Audio Sink y Propiedades.



Nota. Elaborador por el autor.

En el resultado de la **Figura. 82**, la frecuencia central con la que trabaja la recepción de la señal es de 105.3 MHz y para la demodulación del audio el canal selector se ajusta a 104.3 MHz, variando u estableciendo la ganancia del audio en 5.6 dB. Además, se puede observar en la gráfica que se utiliza la Transformada Rápida de Fourier (FFT) para obtener las componentes espectrales de la señal de audio recibida, donde se puede observar que la potencia de cada frecuencia con la que está conformada el audio oscila entre -35 a -100 dB.

Figura. 82 Resultado de recepción en la frecuencia de 104.3Mhz.

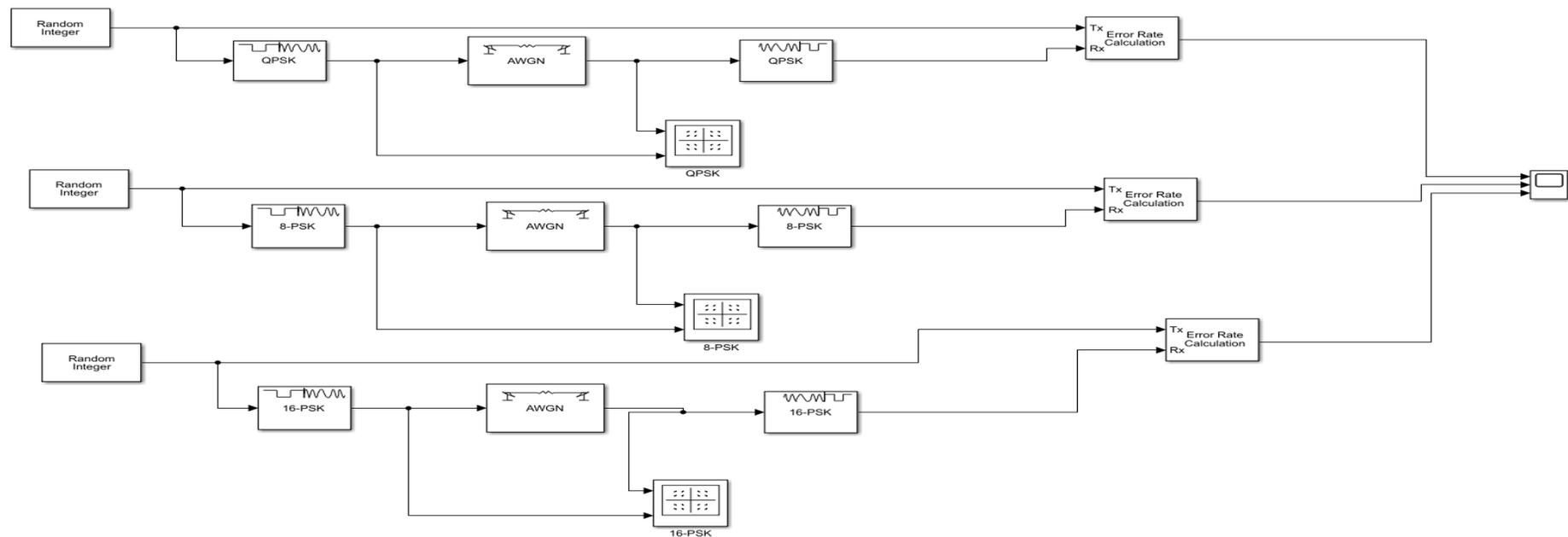


Nota. Elaborador por el autor.

3.6.5 Modulaciones PSK en Simulink

En el siguiente esquema gráfico del programa Matlab se utiliza la herramienta simulink, en la cual proporciona un entorno gráfico mediante bloques de código, en la **Figura. 83** que se presenta a continuación se explora la implementación de las modulaciones 4,8,16-PSK. Donde facilita la comprensión de los principios detrás de la modulación PSK en sistemas de comunicación digital.

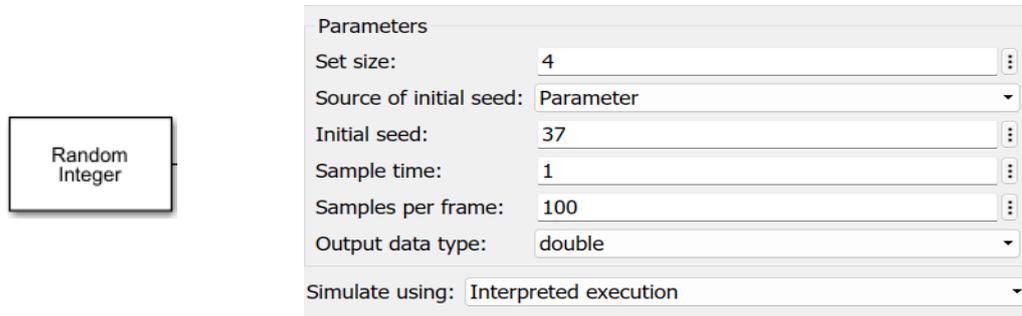
Figura. 83 Modulaciones 4,8,16-PSK en Simulink.



Nota. Elaborador por el autor.

Random Integer: Genera números enteros aleatorios, como observa en la **Figura. 84** que van de 0 a 1.

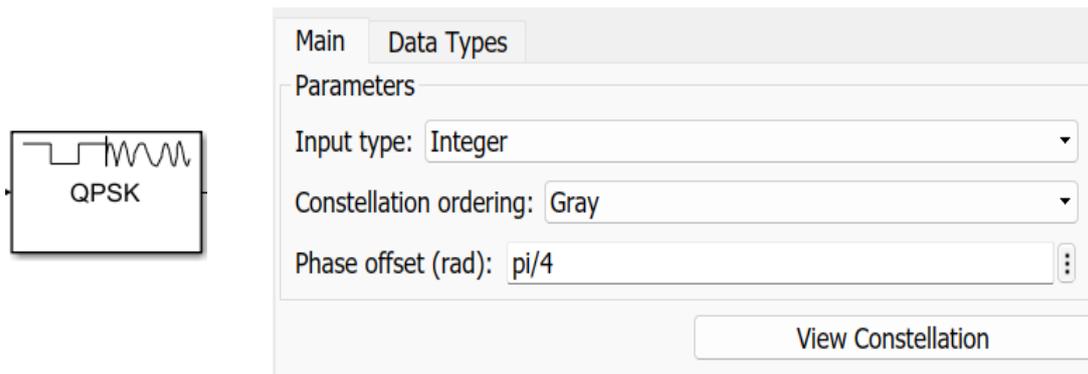
Figura. 84 Bloque *Random Integer* y *Parámetros*.



Nota. Elaborador por el autor.

QPSK Modulador: El bloque que se visualiza en la **Figura. 85** se lo utiliza para modular una señal digital en una señal QPSK.

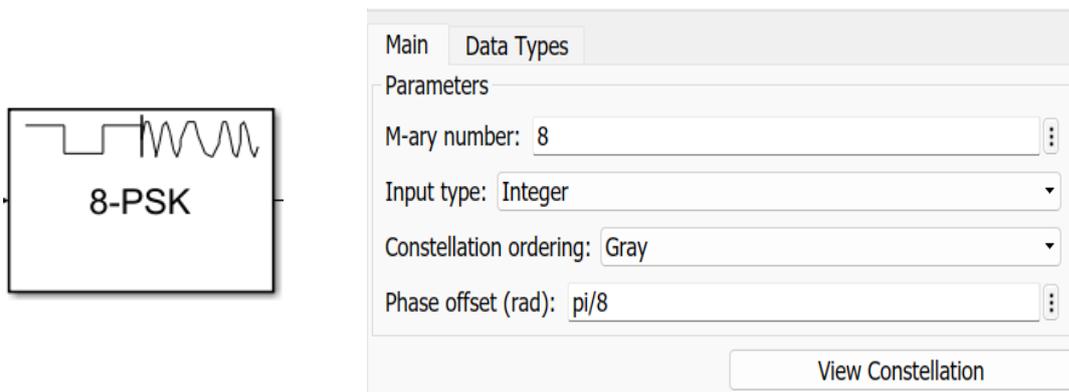
Figura. 85 Bloque *QPSK Modulador* y *Parámetros*.



Nota. Elaborador por el autor.

M-PSK Modulador: El bloque sirve para modular una señal digital en una señal M-PSK, en la cual se puede modificar en valor de M-ary como se muestra en **Figura. 86** que se lo utiliza para modulación 8-PSK, teniendo en cuenta que también puede trabajar con otros valores de modulaciones como lo es 16-PSK.

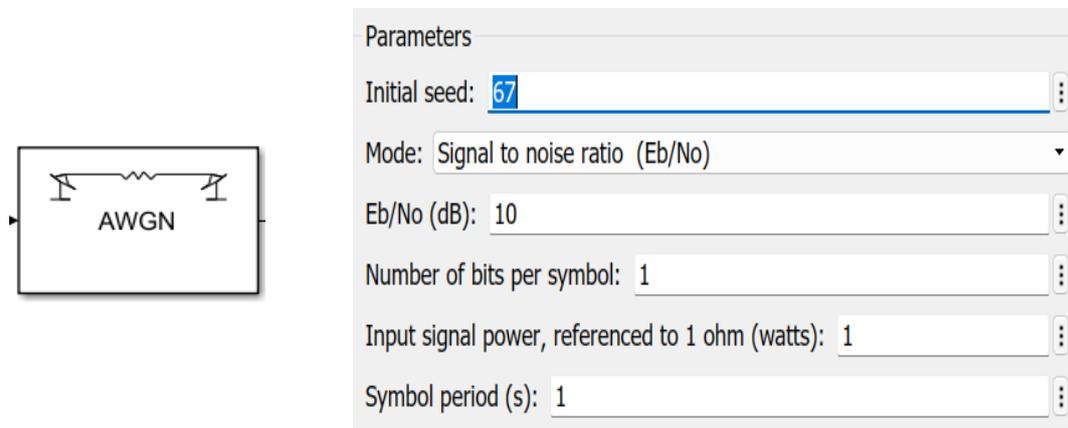
Figura. 86 Bloque 8-PSK Modulador y Parámetros.



Nota. Elaborador por el autor.

AWGN: Este bloque genera la adición del ruido gaussiano en el canal de salida.

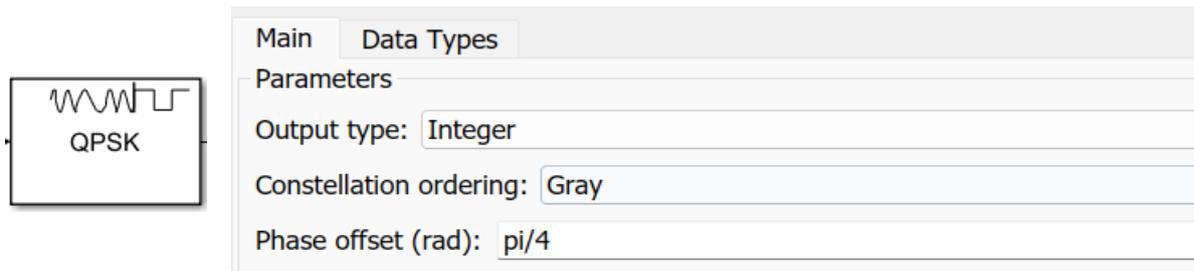
Figura. 87 Bloque AWGN y Parámetros.



Nota. Elaborador por el autor.

QPSK Demodulador: El bloque que se observa en la **Figura. 88** recibe una señal real y la señal de salida es un entero por lo que representa a la constelación en formato de gray, teniendo en cuenta que el inicio o desplazamiento de la onda comienza en $\pi/4$.

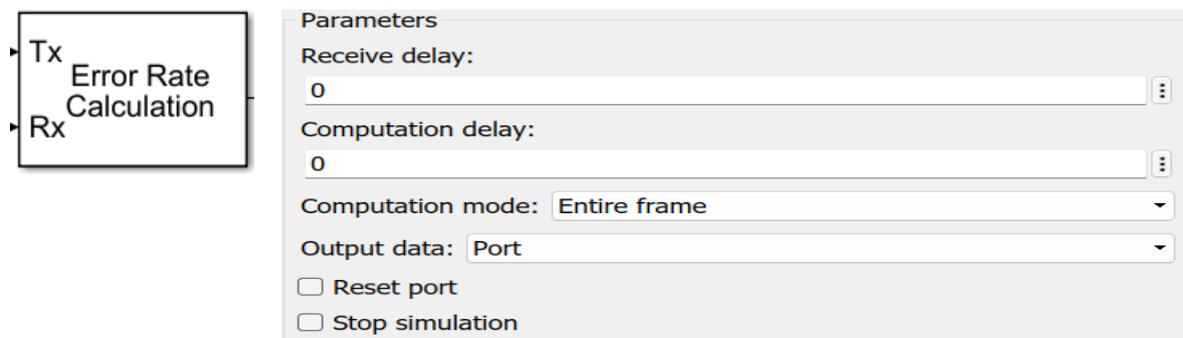
Figura. 88 Bloque QPSK Demodulador y Parámetros.



Nota. Elaborador por el autor.

Error Rate: Se utiliza para calcular en BER en la **Figura. 89** se visualiza los siguientes parámetros: el receiver delay que es el tiempo que llega la señal Tx a la señal Rx y en este caso es 0. El computation delay es el encargado de calcular el BER, observando que en los parámetros de la gráfica este valor es 0.

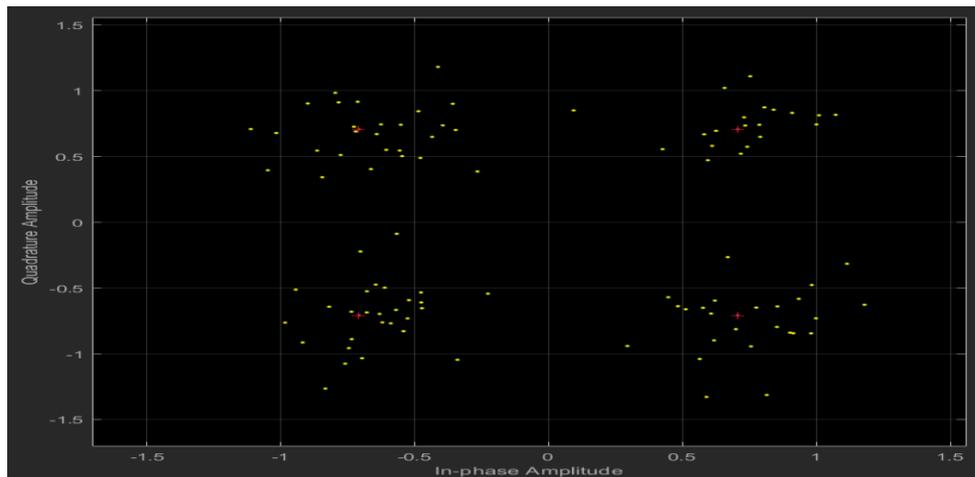
Figura. 89 Bloque Error Rate y Parámetros.



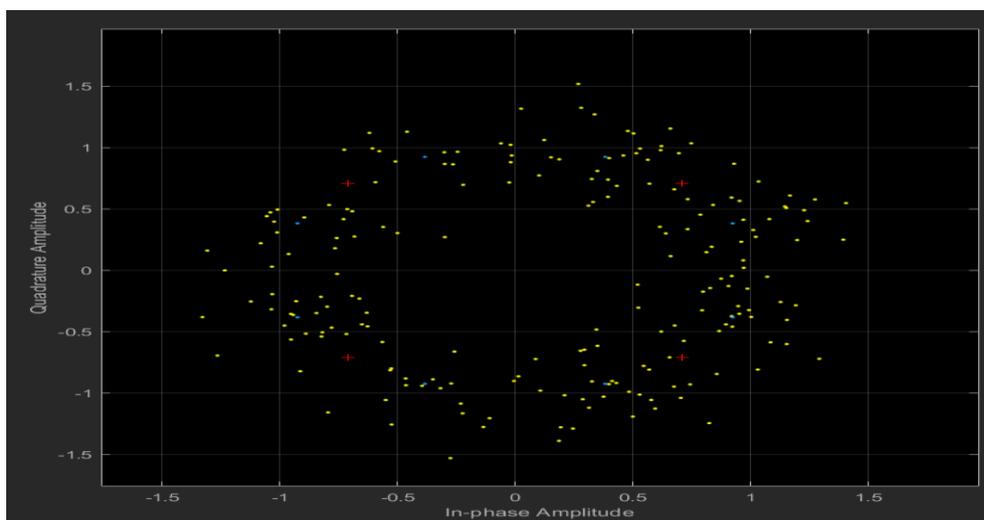
Nota. Elaborador por el autor.

En este caso el bloque "Random Integer" es el que genera la secuencia de datos binarios, que luego se codifica para generar los símbolos y se modula en 4,8,16- PSK como se observa en la **Figura. 90**, los puntos de color azul son las modulaciones ideales. Finalmente, se le agrega un ruido gaussiano que va a afectar con $E_b/N_o = 10\text{dB}$, y se puede apreciar en la **Figura. 90** de los resultados son puntos de dispersión color amarillo.

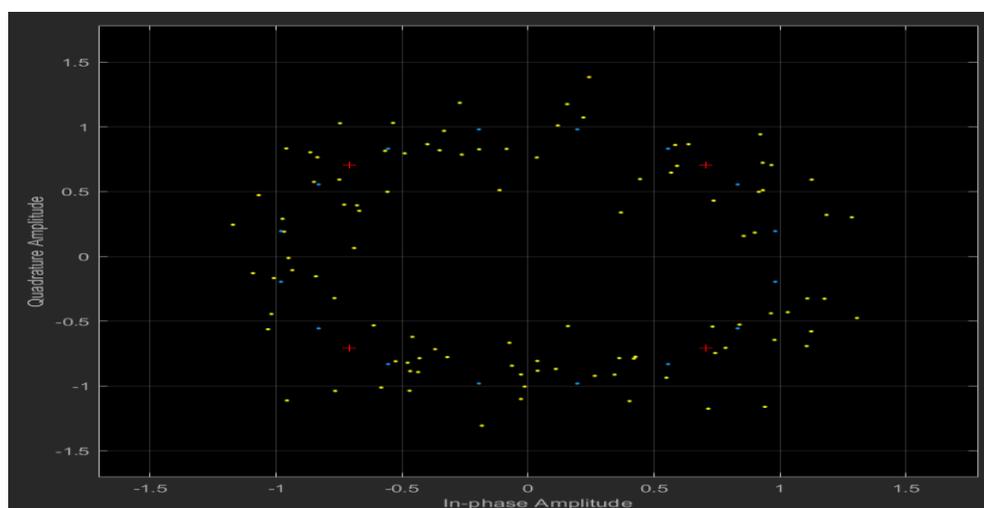
Figura. 90 Diagrama de constelación a) 4-PSK. b) 8-PSK. c) 16-PSK con $E_b/N_o = 10\text{dB}$.



a)



b)



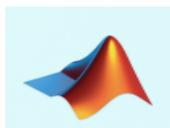
c)

Nota. Elaborador por el autor.

3.6.6 Modulaciones QAM en Guide

En **Figura. 91 a)** se observa el icono del programa Matlab donde se va a utilizar la herramienta guide, que se genera mediante el **Command window** escribiendo **”guide”** como se observa en la **Figura. 91 b)**, se procede a apstar la tecla **Enter**, presento la venta de guide donde se crea la interfaz donde se va a trabajar.

Figura. 91 a) Icono de MATLAB. b) Command window.



a)

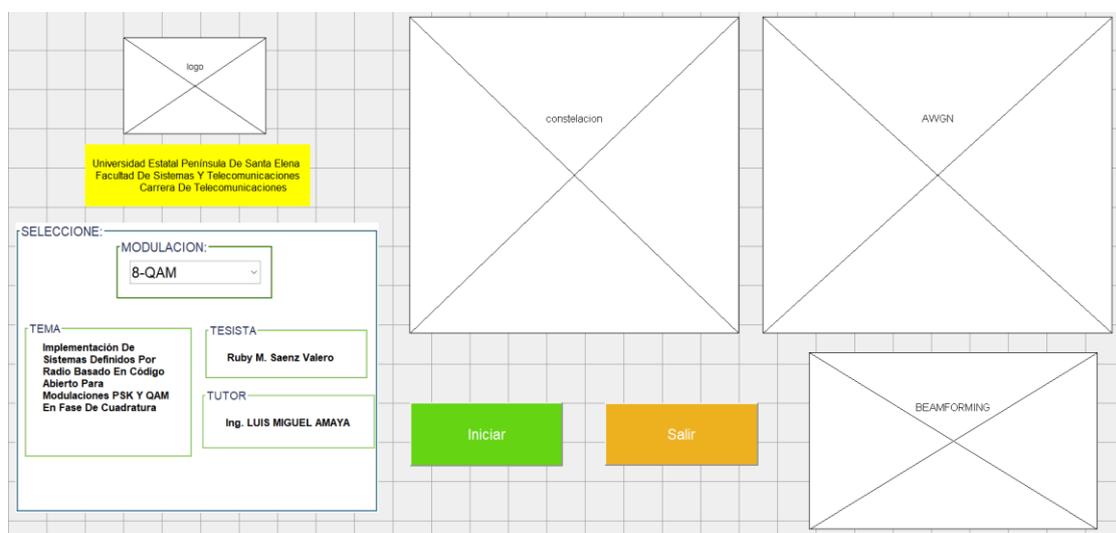


b)

Nota. Elaborador por el autor.

La **Figura. 92** muestra la interfaz gráfica interactiva que permite una comprensión visual y práctica con controles para seleccionar las modulaciones de 8,16,32,64-QAM con sus respectivas gráficas de constelación, ruido gaussiano y tasa de error de bit.

Figura. 92 Ventana de guide.



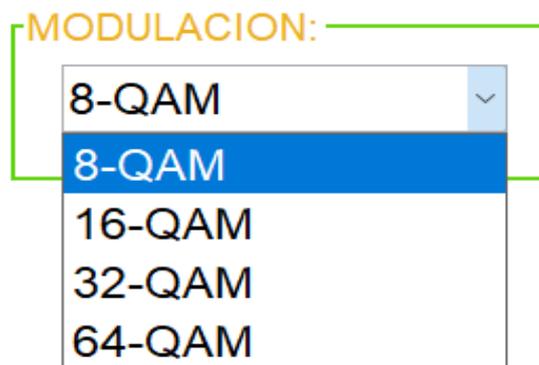
Nota. Elaborador por el autor.

La **Figura. 93**, se observa las modulaciones con las que se va a trabajar y a su vez presenta el código donde se define la función "select_code_Callback", teniendo en cuenta que este se ejecuta cuando se realiza un cambio de valor en las modulaciones que se presenta en el cuadro "select_code", que en la interfaz del usuario se presenta como "Modulación".

Según el valor del caso seleccionado, se actualiza la variable global "M_QAM" para almacenar el valor del código QAM correspondiente (8, 16, 32 o 64). En el interior de la variable se encuentra "handles.cod", que sirve para seleccionar el tipo de caso con el que se va a trabajar validándolo mediante una declaración "switch", el que verifica en qué caso se trabaja.

Por otra parte, en cada caso se cambia la propiedad de los objetos presentados, en los gráficos de la interfaz del usuario a "visible" o "no visible" con la declaración "Set". Por ejemplo, si se selecciona el caso 1, se muestran todos los paneles y se ocultan otros objetos gráficos.

Figura. 93 Selector de modulaciones y codificación.



```
% --- Executes on selection change in select_code.
function select_code_Callback(hObject, eventdata, handles) %#ok<*INUSL,*DEFNU>
global M_QAM %#ok<*NUSED>
handles.cod=get(hObject,'Value');
switch handles.cod
case 1
    M_QAM=8;
    set(handles.banda,'visible','off');
    set(handles.canales,'visible','off');
    set(handles.datarate,'visible','off');
    set(handles.uibuttongroup4,'visible','off');
    set(handles.uibuttongroup5,'visible','off');
    set(handles.uibuttongroup6,'visible','off');
    set(handles.uibuttongroup7,'visible','off');
```

```

set(handles.uipanel1,'visible','on');
set(handles.uipanel2,'visible','on');
set(handles.uipanel3,'visible','on');
case 2
M_QAM=16;
set(handles.uipanel1,'visible','on');
set(handles.uipanel2,'visible','on');
set(handles.uipanel3,'visible','on');
set(handles.banda,'visible','off');
set(handles.canales,'visible','off');
set(handles.datarate,'visible','off');
set(handles.uibuttongroup4,'visible','off');
set(handles.uibuttongroup5,'visible','off');
set(handles.uibuttongroup6,'visible','off');
set(handles.uibuttongroup7,'visible','off');
case 3
M_QAM=32;
set(handles.uipanel2,'visible','on');
set(handles.uipanel3,'visible','on');
set(handles.uipanel1,'visible','on');
set(handles.banda,'visible','off');
set(handles.canales,'visible','off');
set(handles.datarate,'visible','off');
set(handles.uibuttongroup4,'visible','off');
set(handles.uibuttongroup5,'visible','off');
set(handles.uibuttongroup6,'visible','off');
set(handles.uibuttongroup7,'visible','off');
case 4
M_QAM=64;
set(handles.uipanel3,'visible','off');
set(handles.uipanel2,'visible','off');
set(handles.uipanel1,'visible','off');
set(handles.banda,'visible','on');
set(handles.canales,'visible','on');
set(handles.trama,'visible','on');
set(handles.datarate,'visible','on');
set(handles.uibuttongroup4,'visible','on');
set(handles.uibuttongroup5,'visible','on');
set(handles.uibuttongroup6,'visible','on');
set(handles.uibuttongroup7,'visible','on');
end

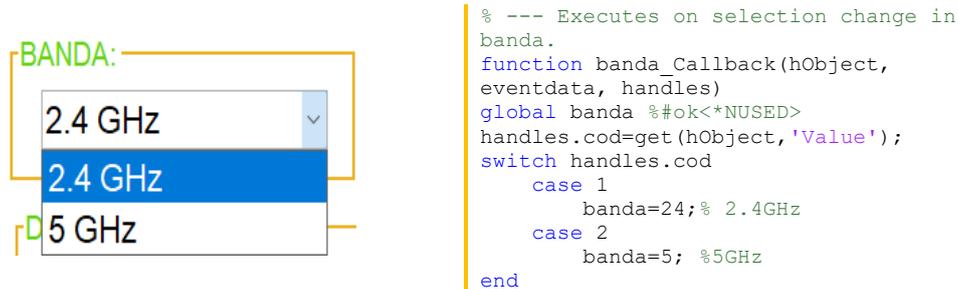
```

Nota. Elaborador por el autor.

Se puede notar claramente que esta selección de opciones de la **Figura. 94**, aparece en la interfaz del usuario cuando se escoge la modulación de 64-QAM. El selector llamado "banda", está conformado de 2 casos y permite a los usuarios seleccionar entre los elementos (2.4 y 5 GHz), también se muestra el código con el que está conformado.

Al comienzo del código, la variable global llamada "banda" cambia según el caso con el que se trabaje, almacenando la selección en "handles.cod" y con la instrucción "switch", que permite verificar en cuál de las dos opciones está trabajando.

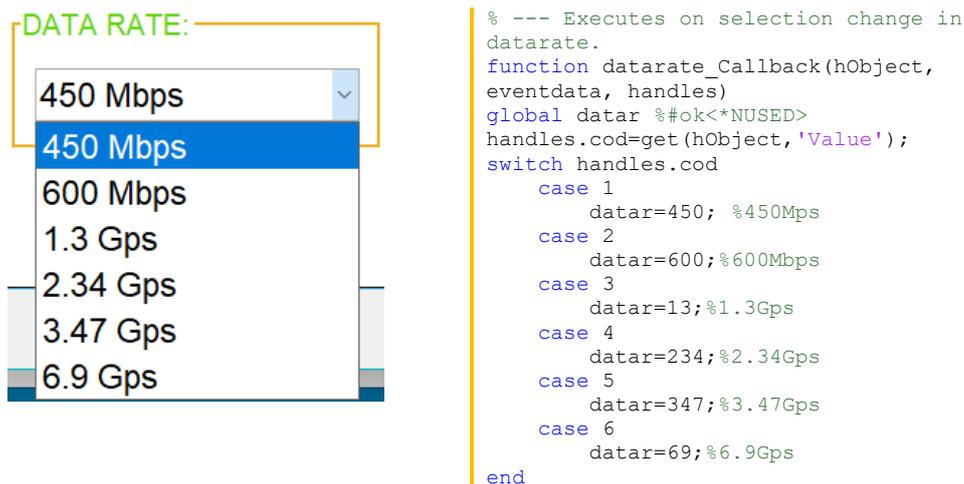
Figura. 94 Selector de Banda y codificación.



Nota. Elaborador por el autor.

En la interface del usuario en la **Figura. 95** se observa el menú desplegable del data rate "tasa de bits" que son los bits transmitidos por el canal. El código se ejecuta mediante la función `datarate_Callback`, con una estructura de `switch-case` que va del caso 1 al 6 cada uno con su respectivo valor.

Figura. 95 Selector de Tasa de bit y codificación.

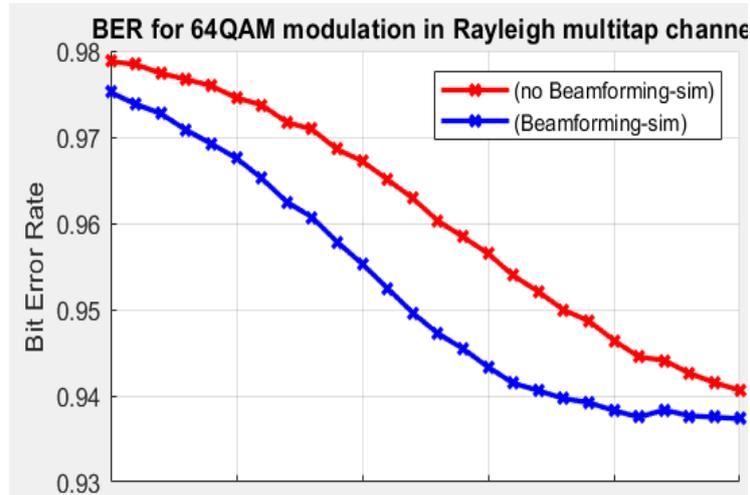


Nota. Elaborador por el autor.

El número de canales que se visualiza en la **Figura. 97 a)**, utiliza la técnica de beamforming que sirva para mejorar la señal - ruido (SNR) recibida en el caso específico de 64-QAM, eliminando las interferencias no deseadas para obtener las gráficas de relación BER

y E_b/N_0 , donde se obtiene una línea roja que representa que no tiene beamforming y la línea azul que si la tiene como se observa en la **Figura. 96**.

Figura. 96 Beamforming.



Nota. Elaborador por el autor.

Por otro lado, en la **Figura. 97 b)**, se visualiza el número de símbolos por cuadro, que es la cantidad de símbolos para la demostración del ruido blanco en el recuadro de las gráficas.

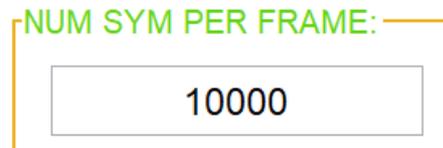
Figura. 97 a) Número de Canales. b) Número de símbolo por trama y codificación.



a)

```
% --- Executes during object creation,
after setting all properties.
function canales_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))

set(hObject, 'BackgroundColor', 'white');
end
```



b)

```
% --- Executes during object creation,
after setting all properties.
function trama_CreateFcn(hObject,
eventdata, handles)
if ispc &&
isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'
))
set(hObject, 'BackgroundColor', 'white');
end
```

Nota. Elaborador por el autor.

El botón iniciar que se muestra en **Figura. 98**, está conformado por el código donde se configuran las técnicas de modulación M-QAM. Es un script destinado para generar diferentes órdenes de modulación (8-QAM, 16-QAM, 32-QAM, 64-QAM).

Es importante notar que este es un análisis general del código. Para un entendimiento más profundo y detalles específicos sobre los cálculos y simulaciones realizadas dentro del código, es necesario revisar con más detalle cada sección del código y las funciones utilizadas en MATLAB y esto se encuentra en el **Anexo 5**.

Figura. 98 Botón Iniciar y codificación.



```
% --- Executes on button press in Iniciar.
function Iniciar_Callback(hObject, eventdata, handles)% Botón iniciar
global M_QAM banda datar
    cla(handles.constelacion,'reset')
    cla(handles.AWGN,'reset')
    cla(handles.BEAMFORMING,'reset')
numSymPerFrame = get(handles.trama,'string') %#ok<*NOPRT,*NASGU>
disp(numSymPerFrame)
numSymPerFrame= str2num(numSymPerFrame) %#ok<*ST2NM>
assignin('base','numSymPerFrame',numSymPerFrame);
Th = get(handles.canales,'string') %#ok<*NOPRT,*NASGU>
Th= str2num(Th) %#ok<*ST2NM>
assignin('base','Th',Th);
if (numSymPerFrame > 0) && (Th > 0)
if (M_QAM==8)
    M = 8; % Modulation order
    x = (0:7); % Integer input
    symbin = qammod(x,M,'bin'); % 16-QAM output (natural-coded binary)
    %symgray = qammod(x,M,'gray'); % 16-QAM output (Gray-coded)
    %scatterplot(symgray,1,0,'b*');
    scatterplot(symbin,1,0,'b. ');
    for k = 1:M
    if k>0 && k<11
        text(real(symbin(k))-0.25, imag(symbin(k))-0.7,...
            num2str(x(k)), 'FontSize',7.5);
    elseif k>10 && k<101
        text(real(symbin(k))-0.5, imag(symbin(k))-0.7,...
            num2str(x(k)), 'FontSize',7.5);
    elseif k>100
        text(real(symbin(k))-0.8, imag(symbin(k))-0.7,...
            num2str(x(k)), 'FontSize',7.5);
    end
    end
%-----
if (M_QAM==32)
    M = 32;
    x = (0:31);
    symbin = qammod(x,M,'bin');
    scatterplot(symbin,1,0,'b. ');
    for k = 1:M
```

```

    if k>0 && k<11
        text(real(symbin(k))-0.25, imag(symbin(k))-0.7,...
            num2str(x(k)), 'FontSize', 7.5);
    elseif k>10 && k<101
        text(real(symbin(k))-0.5, imag(symbin(k))-0.7,...
            num2str(x(k)), 'FontSize', 7.5);
    elseif k>100
        text(real(symbin(k))-0.8, imag(symbin(k))-0.7,...
            num2str(x(k)), 'FontSize', 7.5);
    end
end
axis([-6 6 -6 6])
saveas(gcf, 'Constelacion.png')
hold off;
title('8-QAM ')
axes(handles.constelacion);
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;

end
%*****Modulación de amplitud en cuadratura M-QAM*****
M = 64;
k = log2 (M);
data = randi([0 1], numSymPerFrame*k, 1);
txSig = qammod(data, M, 'InputType', 'bit', 'UnitAveragePower', true);
snrdB = 25;
rxSig = awgn (txSig, snrdB, 'measured');
%***** TRAZO DE DIAGRAMA DE CONSTELACION *****
hold on;
scatterplot(txSig);
saveas(gcf, 'Constelacion.png')
hold off;
title('64-QAM ')
axes(handles.constelacion);
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;
scatterplot(rxSig); %#ok<*NASGU>
saveas(gcf, 'AWGN.png')
hold on
grid
hold off
axes(handles.AWGN);
path = 'AWGN.png';
img = imread(path);
imshow(img);
axis off;
%***** BER Beamforming usando modulación M-QAM *****
N = 9^6;
k = sqrt(1/((2/3)*(M-1)));
m = [1:sqrt(M)/2]; %#ok<*NBRAK>
alphaMqam = [-(2*m-1) 2*m-1];
Eb_N0_dB = [0:25];
ipHat = zeros(1, N);
for ii = 1:length(Eb_N0_dB)
    ip = randsrc(1, N, alphaMqam) + j*randsrc(1, N, alphaMqam);
    s = k*ip;
    n = 1/sqrt(2)*[randn(1, N) + j*randn(1, N)];
    h = 1/sqrt(2)*[randn(Th, N) + j*randn(Th, N)];
    signal = (1/sqrt(Th))*kron(ones(Th, 1), s);
    phase = exp(j*angle(h));
    phase_conjugate = conj(phase);
    bf_ch = h.*phase_conjugate;
    y1 = sum(h.*signal, 1) + 10^(-Eb_N0_dB(ii)/20)*n;
    y2 = sum(bf_ch.*signal, 1) + 10^(-Eb_N0_dB(ii)/20)*n;
    y1Hat = y1./sum(h, 1);
    y2Hat = y2./sum(bf_ch, 1);
    y_re1 = real(y1Hat);
    y_im1 = imag(y1Hat);
    y_re2 = real(y2Hat);
    y_im2 = imag(y2Hat);
    ipHat_re1 = 2*floor(y_re1/2)+1;
    ipHat_re1(find(ipHat_re1>max(alphaMqam))) = max(alphaMqam);
    ipHat_re1(find(ipHat_re1<min(alphaMqam))) = min(alphaMqam);
    ipHat_re2 = 2*floor(y_re2/2)+1;

```

```

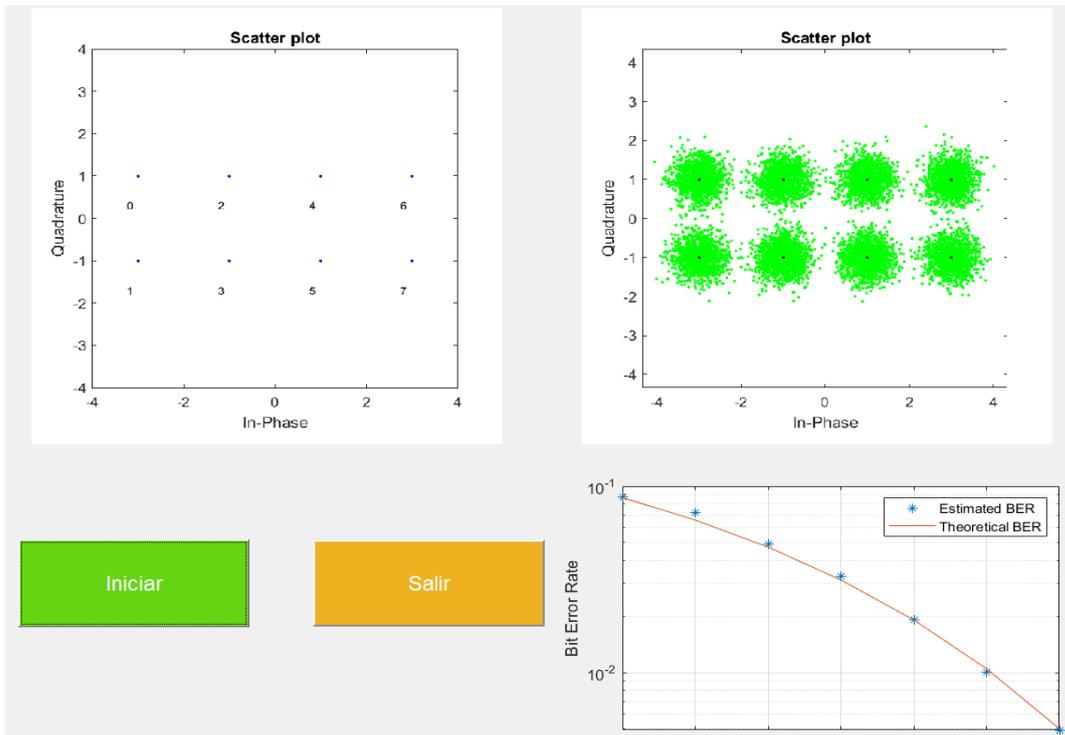
ipHat_re2(find(ipHat_re2>max(alphaMqam))) = max(alphaMqam);
ipHat_re2(find(ipHat_re2<min(alphaMqam))) = min(alphaMqam);
ipHat_im1 = 2*floor(y_im1/2)+1;
ipHat_im1(find(ipHat_im1>max(alphaMqam))) = max(alphaMqam);
ipHat_im1(find(ipHat_im1<min(alphaMqam))) = min(alphaMqam);
ipHat_im2 = 2*floor(y_im2/2)+1;
ipHat_im2(find(ipHat_im2>max(alphaMqam))) = max(alphaMqam);
ipHat_im2(find(ipHat_im2<min(alphaMqam))) = min(alphaMqam);
ipHat1 = ipHat_re1 + j*ipHat_im1;
nErr1(ii) = size(find([ip- ipHat1]),2);
ipHat2 = ipHat_re2+ j*ipHat_im2;
nErr2(ii) = size(find([ip- ipHat2]),2);
end
simBer1 = nErr1/N;
simBer2 = nErr2/N;
axes(handles.BEAMFORMING);%Axis BEAMFORMING donde se realizará la gráfica en axis
semilogy(Eb_N0_dB,simBer1,'rx-','LineWidth',2);
hold on
semilogy(Eb_N0_dB,simBer2,'bx-','LineWidth',2);
axis auto
grid on
legend(' (no Beamforming-sim) ',' (Beamforming-sim) ');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BER for 64QAM modulation in Rayleigh multitap channel');
msgbox('"Beamforming Technique implemented Successfully!"','Result');
%***** TASA DE ERROR DE SIMBOLO (SER) *****
y = gammod(data,M,'bin','InputType','bit','OutputDataType', ...
    numerictype(1,M,10));
demodData = qamdemod(y,M,'bin','OutputType','bit');
[numErrors1,ser1] = symerr(data,demodData)
demodData = qamdemod(txSig,M,'bin','OutputType','bit');
[numErrors,ser] = symerr(data,demodData)
elseif (M_QAM==128) && (banda==5) && (datar==13)
%-----
    % hold off;
    cla(handles.constelacion,'reset')
    cla(handles.AWGN,'reset')
    cla(handles.BEAMFORMING,'reset')
    ipHat2 = ipHat_re2+ j*ipHat_im2;
    nErr2(ii) = size(find([ip- ipHat2]),2);
end
simBer1 = nErr1/N;
simBer2 = nErr2/N;
%***** TASA DE ERROR DE SIMBOLO (SER) *****
y = gammod(data,M,'bin','InputType','bit','OutputDataType', ...
    numerictype(1,M,10));
demodData = qamdemod(y,M,'bin','OutputType','bit');
[numErrors1,ser1] = symerr(data,demodData)
demodData = qamdemod(txSig,M,'bin','OutputType','bit');
[numErrors,ser] = symerr(data,demodData)
else
    errordlg(';Error Por favor revise la tabla e intente nuevamente!','ERROR DE
    SELECCIÓN ')
end
end
else
    errordlg(';Error El número de canales y simbolos debe ser mayor a cero!','ERROR
    NÚMERO DE CH-SYM ')
end

```

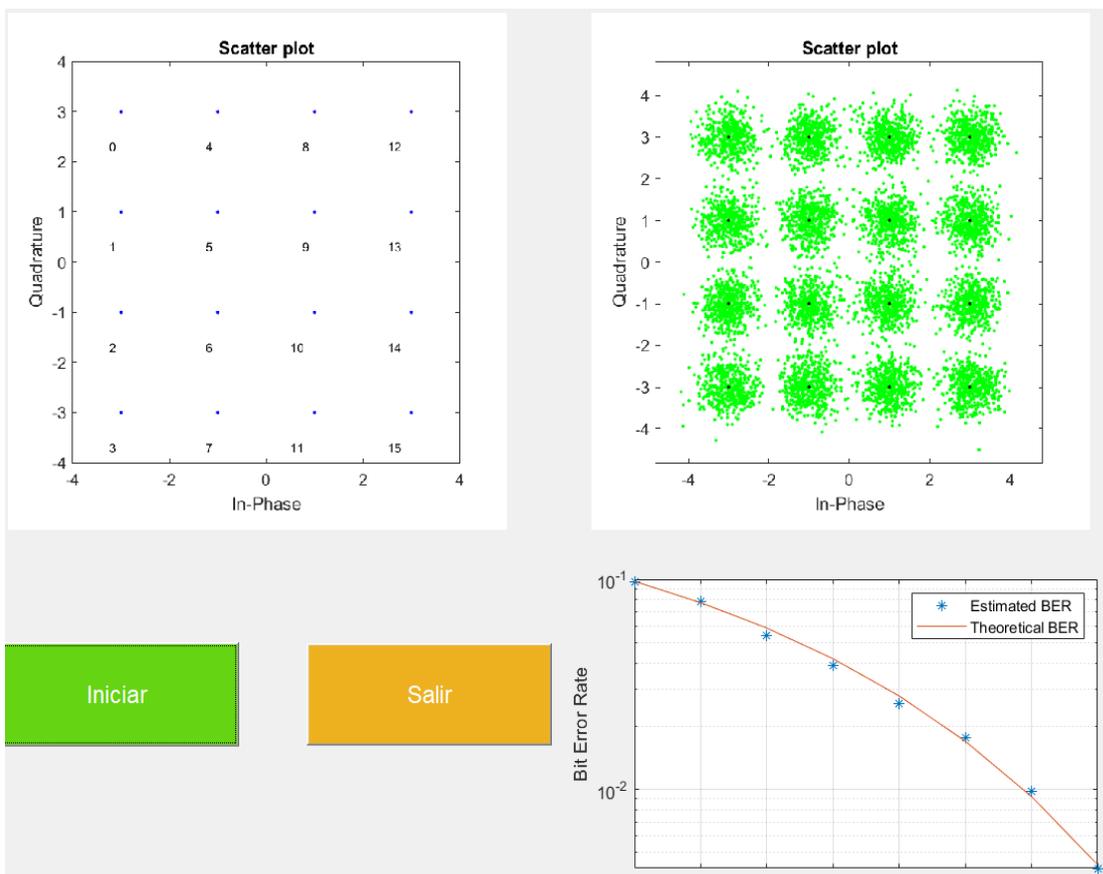
Nota. Elaborador por el autor.

A continuación, al ejecutar el código que se mostró en la **Figura. 98**, se realiza la simulación para cada caso, generando un diagrama de constelación, introduciendo ruido (AWGN) en la señal, y calculando la tasa de error de bit (BER) y tasa de error de símbolo (SER) como se presenta en el resultado de la **Figura. 99**.

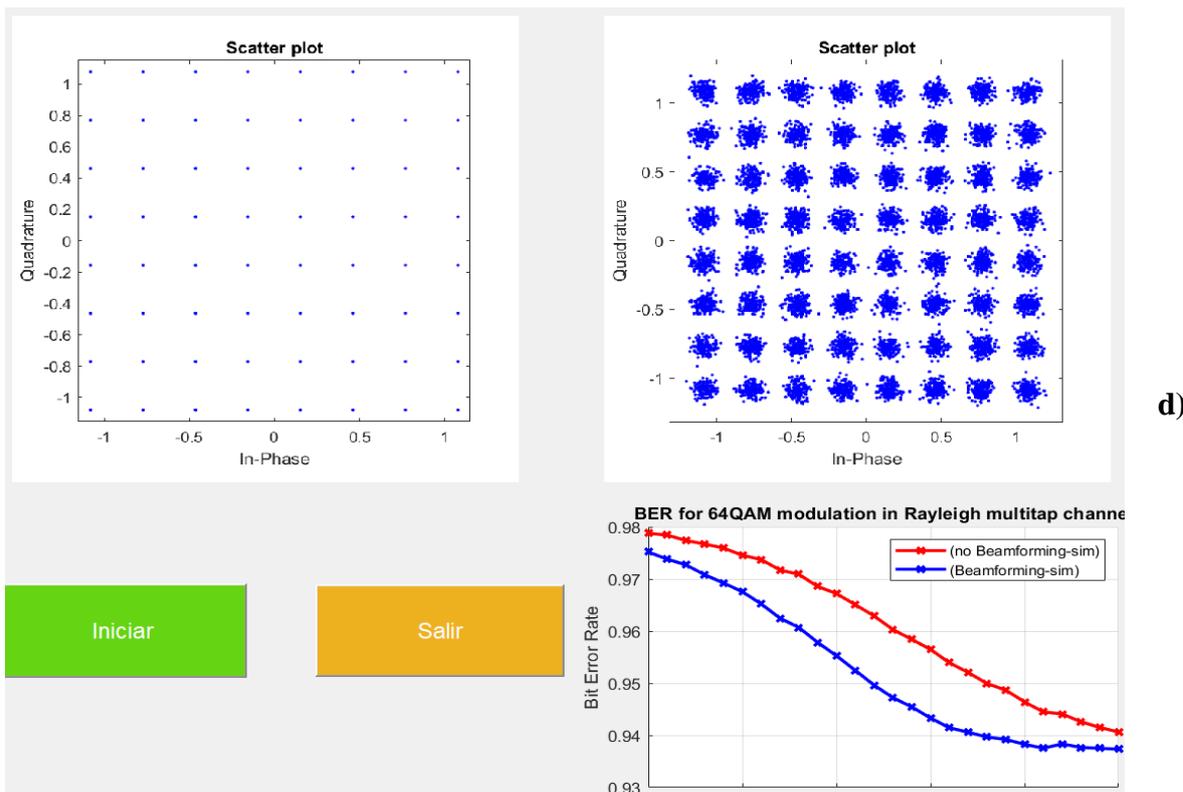
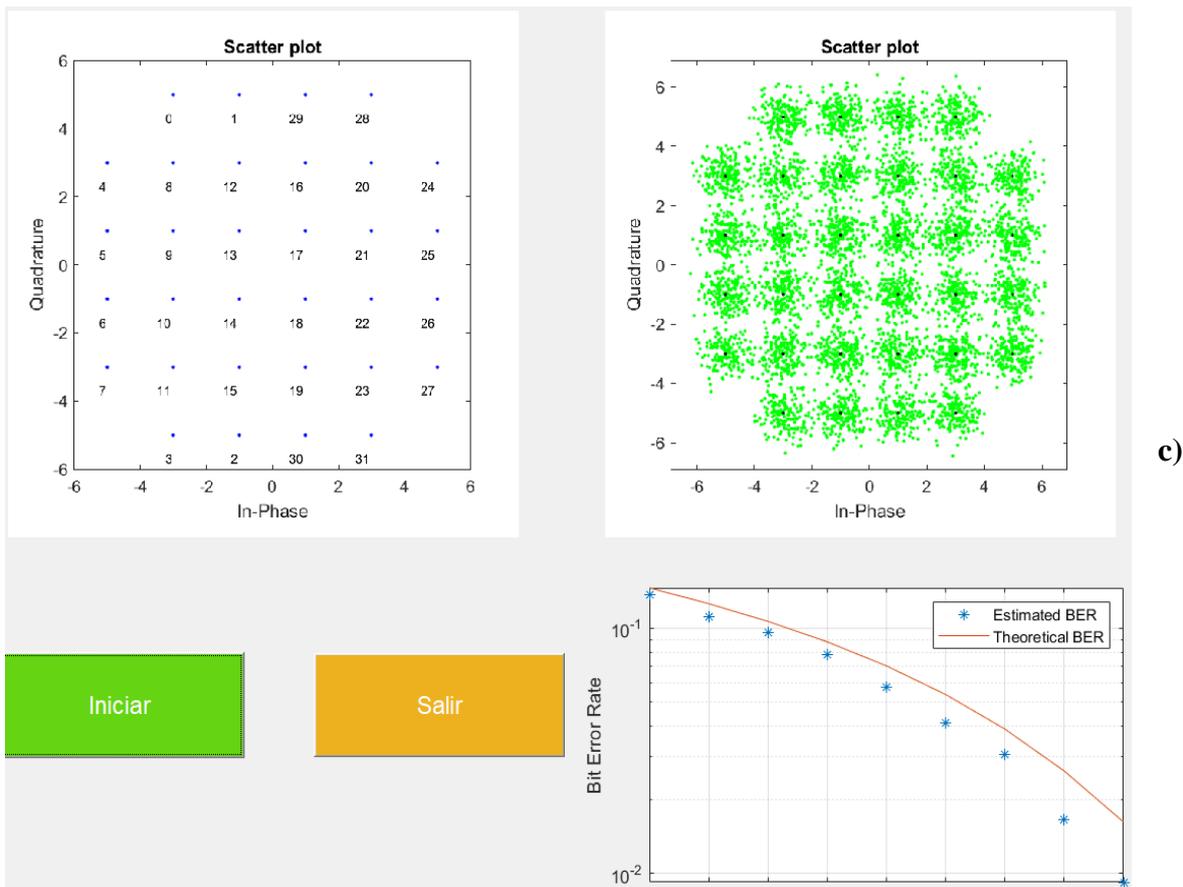
Figura. 99 Resultados se las modulaciones a) 8-QAM. b) 16-QAM. c) 32-QAM. d) 64-QAM en Guide Matlab.



a)



b)



Nota. Elaborador por el autor.

Cuando se da clic el botón salir de la **Figura. 100**, aparece una ventana donde consulta al usuario "¿Desea salir del programa? " con las opciones (si o no), teniendo en cuenta que si la opción seleccionada fue "No" la función finaliza y no se va a realizar ningún tipo de cambio, si la opción fue "Si" se borran todas las gráfica de la interfaz del usuario y se cierra la ventana.

Figura. 100 Botón Salir y codificación.



```
% --- Executes on button press in Salir.  
function Salir_Callback(hObject, eventdata, handles)  
a = questdlg('¿Desea salir del programa?', 'Salir', 'Si', 'No', 'No');  
if strcmp(a, 'No')  
    return;  
end  
clc;  
clear all; %#ok<*CLALL>  
close all
```

Nota. Elaborador por el autor.

CAPÍTULO 4: Análisis de Resultados, Conclusión Y Recomendación.

4.1 Análisis De Los Resultados

Durante el desarrollo de este proyecto, se mostró diferentes gráficas del diagramas de constelación según la modulación, ya sea PSK o QAM, también se explicó cómo se efectuaron las mediciones y gráficas en los resultados.

Los diagramas de constelación representan gráficas bidimensional complejas, ya que, presenta interferencia y distorsión que afectan a la transmisión, esto se puede justificar mediante la relación señal-ruido (SNR) obtenida.

Para analizar los resultados obtenidos se utilizó el método de aprendizaje visual, que se basa en la observación e interpretación de gráficos como método principal, analizando los diagramas de constelación generados por Gnu Radio y MATLAB (Guide-PSK, Simulink-QAM), también el espectro de frecuencia del receptor originado por Gnu Radio.

En comparación de los resultados de las modulaciones de 4, 8, 16- PSK generadas por Gnu Radio de la **Figura. 61**, se puede observar que en todas las modulaciones PSK, las constelaciones se encuentran difusas o borrosas esto se debe al ruido presentado en la transmisión debido a diversos factores tales como: interferencia de electromagnética, ruido en el canal de transmisión proveniente de dispositivos electrónicos cercanos, atenuación de la señal o problemas en la modulación de la señal. En la **Figura. 61 a)**, se puede ver claramente la modulación 4-PSK a diferencia de la **Figura. 61 b) y c)**, que casi no se distinguen las constelaciones 8, 16 -PSK, al igual que en la **Figura. 90 b) y c)**, que se generaron en Guide (Matlab), ya que se presenta un ruido 10dB y en la **Figura. 90 a)**, muestra un cambio de fase de la señal modulada, además se visualizan las modulaciones ideales en color azul. Estos resultados se pueden respaldar de manera cuantitativa en base a los cálculos obtenidos de la

SNR de 4,8 y 16 PSK donde la modulación de 4 PSK presenta una relación señal-ruido (SNR) de 4.10 dB siendo mejor en comparación a la SNR de 8 y 16 PSK las cuales son 0.49 dB y 0.2 dB respectivamente, de esta manera se respalda que esta modulación es más tolerante al ruido en el sistema de transmisión utilizado.

Al analizar los resultados de las modulaciones de 16, 32, 64-QAM, en la **Figura. 68 a)** y **b)**, se transmite en la frecuencia de 104.3 MHz, con una ganancia de transmisión de 20 dB. En el caso de la modulación de 16-QAM en la **Figura. 68 a)**, se visualiza la constelación con 16 agrupaciones de los puntos de dispersión acompañado de ruido de transmisión, al igual que la modulación de 64-QAM de la **Figura. 68 b)**, el tamaño de la constelación presentada es mayor en comparación a 16-QAM, lo que indica que la intensidad es alta. Los resultados cuantitativos respaldan la afirmación mediante los cálculos de la relación señal-ruido (SNR) de 16 y 64 QAM. Se observa que la modulación 16 QAM exhibe una SNR de 3.24 dB, superando la SNR de 64 QAM, que es de -0.40 dB. Esto respalda la idea de que la modulación 16 QAM es más resistente al ruido en el sistema de transmisión empleado, en comparación con la modulación 64 QAM.

En la **Figura. 75**, se ha creado el diagrama de constelación generado en Gnu Radio de 32-QAM ideal, que representa los datos transmitidos y recibidos ya que, al mostrar la modulación sus puntos son uniformes y no presentan ningún tipo de interferencia o ruido.

Por otro lado, en la **Figura. 99**, se encuentra las modulaciones de 8, 16, 32 y 64 QAM que se generó mediante el script que se detalla en el **Anexo 5**, como resultado genera tres gráficas por modulación la primera presenta la constelación ideal sin ningún tipo de ruido o distorsión, en la segunda imagen representa la constelación con tipo de ruido gaussiano y en la tercera gráfica se muestra la tasa de error de bit, el

comportamiento de la gráfica se basa en la medida que aumenta la subportadora en BER disminuye lo que da a entender que tiene una mayor tolerancia a la interferencia o ruido.

Por último, se encuentra la **Figura. 82**, el receptor de la señal, en la gráfica de la transformada rápida de Fourier (FFT Plot) que permitiendo descomponer la señal espectral, este proporcionando dos ejes: el eje vertical que es la amplitud trabajando en dB y el eje horizontal que representa a la frecuencia en MHz, como se visualiza en la gráfica se trabaja con una frecuencia central de 105.3 MHz y ajustando la señal del canal en 104.3 MHz, en la cual permite ajustar la ganancia el audio mejorando la calidad del audio.

4.2 Conclusión

En conclusión, se ha logrado exitosamente todos los objetivos planteados. A través de la identificación de los conceptos clave, que ha alcanzado conocimientos sólidos de los fundamentos teóricos sobre las modulaciones PSK y QAM. Explicando todos los recursos necesarios en el marco conceptual siendo una herramienta importante que respalda el aprendizaje de los conceptos base.

En definitiva, el análisis de las técnicas de las modulaciones ha permitido determinar las variables adecuadas para el diseño de los bloques lógicos en Gnu Radio, vale decir, que se ha seleccionado trabajar con el equipo SDR HackRF One, ya que es el adecuado y se ajusta a las necesidades del proyecto tanto en características y presupuesto, facilitando llevar a cabo el uso del equipo HackRF One con la compatibilidad del software Gnu radio y dando paso a la construcción del módulo educativo.

Finalmente, se realizó la guía didáctica dando una explicación exhaustiva del funcionamiento de los bloques lógicos y las simulaciones en el desarrollo de la propuesta, permitido que los estudiantes de las asignaturas de comunicaciones digitales

y procesamiento digital de señales de la carrera de telecomunicaciones, se les facilite la comprensión y aplicación de estas modulaciones con el equipo SDR HackRF One.

4.3 Recomendación

- Para asegurar una transmisión de audio sin interferencias, se recomienda apagar los dispositivos electrónicos, ya que su funcionamiento puede perturbar la señal de radiofrecuencia debido a las ondas electromagnéticas que emanan.
- En situaciones donde el filtro empleado no garantice la óptima recepción de la señal, se aconseja la implementación de bloques complementarios destinados a la recuperación de la señal con la menor tasa de errores posible. Este enfoque busca asegurar la fiabilidad y calidad en la recepción de la señal, incluso en escenarios donde el filtro inicial pueda resultar insuficiente.
- Se sugiere la utilización del equipo USRP para la exploración y aplicación de modulaciones avanzadas. Asimismo, se recomienda la implementación de otras prácticas complementarias para enriquecer el estudio y análisis de estas modulaciones de alta complejidad.

Materiales De Referencias

Bibliografía

- 1] | The MathWorks, Inc., «Mathworks.com,» 1994 - 2022. [En línea]. Available:
1] <https://www.mathworks.com/products/matlab.html>. [Último acceso: 26 Junio 2022].
- 2] | Proyecto 2022 GNU Radio, «GNU Radio,» 2022. [En línea]. Available:
2] <https://www.gnuradio.org/about/>. [Último acceso: 26 junio 2022].
- 3] | Greatscottgadgets, «Greatscottgadgets.com,» 2009 - 2021. [En línea].
3] Available: <https://greatscottgadgets.com/hackrf/one/>. [Último acceso: 27 junio 2022].
- 4] | RAE, « Real Academia Española,» [En línea]. Available:
4] <https://dle.rae.es/telecomunicaci%C3%B3n>. [Último acceso: 11 Junio 2022].
- 5] | E. Ortega, La comunicación publicitaria, Madrid : Pirámide, 2004, 2009.
- 6] | Moragas, «Sociología de la comunicación de masas,» *Les Cahiers de la
6] Communication*, vol. 1, nº 1, pp. 18 - 36, 1981.
- 7] | G. D. F. VILLAFANA, «FUNDAMENTOS DE
7] TELECOMUNICACIONES,» [En línea]. Available:
[https://www.itmerida.mx/panel/posgrado/archivos/mi/Fundamentos%20de%20Telecomunicaciones%20Unidad%201%20\(1\).pdf](https://www.itmerida.mx/panel/posgrado/archivos/mi/Fundamentos%20de%20Telecomunicaciones%20Unidad%201%20(1).pdf). [Último acceso: 20 mayo 2022].
- 8] | «Perturbaciones en la Red - Cableado Estructurado, Fibra Óptica, Redes,
8] Consultoría,» 21 Septiembre 2021. [En línea]. Available:
<https://microinstalaciones.com.ar/perturbaciones-en-la-red/>. [Último acceso: 20
mayo 2022].
- 9] | R. lacayo, «Speakeasy: the military software radio,» *IEEE*, vol. 33, nº 5, pp.
9] 56-61, 1995.

- 10] | D. Nieva y J. Sanabria, Sistema De Comomunicación SDR Para Modulación PSK y QAM, Santiago de Cali , 2018.
- 11] | K. M.-P. Ahmad Ismail, «QAM and PSK modulation performance analysis,» *Energy Sources, Part A: Recovery, Utilization, and*, vol. 44, nº 1, 2019.
- 12] | V. T. BETANCUR, «ALGORITMO DE ESTIMACIÓN DE PARÁMETROS Y MODULACIÓN DE UNA SEÑAL RECIBIDA POR UN SDR,» MEDELLÍN, 2017.
- 13] | O. A. MOLINA, «DISEÑO E IMPLEMENTACIÓN DE UN MÓDULO EFICIENTE MODULADOR/ DEMODULADOR QAM PARA GNU RADIO,» *Gerencia Tecnológica Informática*, vol. 15, nº 43, pp. 63-77, 2016.
- 14] | Universidad Autónoma del Estado de Hidalgo, «Centro de Innovación para el Desarrollo y la Capacitación en Materiales Educativos,» [En línea]. Available: http://cidecame.uaeh.edu.mx/lcc/mapa/proyecto/libro27/38_concepto_y_necesidad_de_modulacin.html. [Último acceso: 10 Noviembre 2023].
- 15] | J. G. C. Santiana, «Universidad Católica De Santiago De Guayaquil,» 30 Octubre 2014. [En línea]. Available: <http://repositorio.ucsg.edu.ec/bitstream/3317/2738/1/T-UCSG-POS-MTEL-28.pdf>. [Último acceso: 21 Agosto 2022].
- 16] | «Caítulo 2: transmisión y recepción,» La Catarina UDLAP, [En línea]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/escobar_a_g/capitulo2.pdf. [Último acceso: 14 Noviembre 2023].

- 17] | «MODULACIONES ESPECIALES,» [En línea]. Available:
<https://catedras.facet.unt.edu.ar/ft/wp-content/uploads/sites/123/2017/03/10-MODULACIONES-ESPECIALES.pdf>. [Último acceso: 11 Noviembre 2023].
- 18] | Isao Matsumoto, «rohm,» 1997 - 2023. [En línea]. Available:
<https://www.rohm.com/electronics-basics/wireless/modulation-methods>. [Último acceso: 10 Noviembre 2023].
- 19] | J. E. S. Rodriguez, «Telecomundo,» Modulaciones Digitales, 16 Octubre 2016. [En línea]. Available: <https://telecomundo.wordpress.com/modulaciones-digitales/>. [Último acceso: 10 Noviembre 2023].
- 20] | J. A. Sills, «Maximum-likelihood Modulation Classification for PSK/QAM,» 1999. [En línea]. Available:
<https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.675.1657>. [Último acceso: 23 Agosto 2022].
- 21] | slideshare, «Modulacion digital,» 23 Junio 2012. [En línea]. Available:
<https://es.slideshare.net/cctobarial/modulacion-digital-13433132>. [Último acceso: 11 Noviembre 2023].
- 22] | G. B. a. A. S. G.S. O. Popescu, «Review of PSK and QAM — Digital modulation techniques on FPGA,» *International Joint Conference on Computational Cybernetics and Technical Informatics*, n° doi: 10.1109/ICCCYB.2010.5491254., pp. pp. 327-332, 2010.
- 23] | B. A. Forouzan, TRANSMISIÓN DE DATOS Y REDES DE COMUNICACIONES 4ED, Madrid: McGraw-Hill, 2011.

- [D. Rosario, «Procesos estocásticos,» pmf – Modulación BPSK, 3 Enero 2018.
24] [En línea]. Available: <http://blog.espol.edu.ec/estg1003/modulacion-bpsk-pmf/>.
[Último acceso: 22 Agosto 2022].
- [«Modulación Digital,» Sistemas de Telecomunicación, [En línea]. Available:
25] <https://www.uv.es/~hertz/hertz/Docencia/teoria/Trasmdigital.pdf>. [Último acceso: 23
Agosto 2022].
- [«Electrónica Fácil,» MODULACIÓN DIGITAL, 2022. [En línea]. Available:
26] [https://www.electronicafacil.net/tutoriales/MODULACION-DIGITAL-FSK-PSK-
QAM.html](https://www.electronicafacil.net/tutoriales/MODULACION-DIGITAL-FSK-PSK-QAM.html). [Último acceso: 23 Agosto 2022].
- [«Modulación por desplazamiento de fase,» 27 Octubre 2023. [En línea].
27] Available:
https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_desplazamiento_de_fase.
[Último acceso: 11 Noviembre 2023].
- [J. L. Rodríguez y S. C. Suero, Modulación de señales digitales, Sevilla:
28] UNIVERSIDAD DE SEVILLA DEPARTAMENTO DE TECNOLOGÍA
ELECTRÓNICA, 1995.
- [H. H. Nguyen y E. Shwedyk, A First Course in Digital Communications,
29] Cambridge, England: Cambridge University Press, 2009.
- [I. H. Rioja, «PROCESADO DE SEÑAL EN COMUNICACIONES,» TEMA
30] 3 : MODULACIÓN QAM , [En línea]. Available:
<https://aholab.ehu.eus/users/inma/psc/tema3.pdf>. [Último acceso: 29 Octubre 2023].

- | «MODULACION DIGITAL,» 2004. [En línea]. Available:
31] <https://www.electronicafacil.net/tutoriales/MODULACION-DIGITAL-FSK-PSK-QAM.html>. [Último acceso: 12 Noviembre 2023].
- | Ian Poole, «Electronics Notes,» [En línea]. Available:
32] <https://www.electronics-notes.com/articles/radio/modulation/quadrature-amplitude-modulation-types-8qam-16qam-32qam-64qam-128qam-256qam.php>. [Último acceso: 18 Noviembre 2023].
- | L. Pardo y E. Fernandez, «Neoteo,» SDR – Radio Definida por Software, 4
33] Mayo 2012. [En línea]. Available: <https://www.neoteo.com/sdr-radio-definida-por-software/>. [Último acceso: 28 Agosto 2022].
- | F. Fernandez, «SDR : EQUIPOS DE RADIO DEFINIDOS POR,» [En línea].
34] Available: <https://www.ea1uro.com/sdr1/sdr.htm>. [Último acceso: 23 Agosto 2022].
- | J. Rodríguez De Haro y A. & Roldán Aranda, Análisis software y hardware
35] del SDR HackRF One, Universidad de Granada, 2017.
- | Hak5, «Hak5,» [En línea]. Available: <https://shop.hak5.org/products/hackrf>.
36] [Último acceso: 24 octubre 2023].
- | D. Sánchez Ibarra, «Diseño, Construcción y Caracterización de antenas para
37] UHF.,» 2007. [En línea]. Available: [http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/sanchez_i_d/capitulo_3.html#](http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/sanchez_i_d/capitulo_3.html#.)
. [Último acceso: 28 Agosto 2022].
- | W. Tomasi, G. Hernández y V. Pozo, «Sistemas de comunicaciones
38] electrónicas,» Pearson Educación, 2003.

- 39] | W. Tomasi, *Sistemas de Comunicaciones Electrónicas* 4ta ed., México: Pearson Education., 2003.
- 40] | B. A. Fette, *Cognitive radio technology*, Academic Press: 1era Edición, 2009.
- 41] | T. F. Collins, R. Getz, D. Pu y A. M. Wyglinski, *Software-Defined Radio for Engineers*, Norwood, MA, Estados Unidos de América: Artech House, 2018.
- 42] | «Ettus Research,» NI. All Rights Reserved, 2022. [En línea]. Available: <https://www.ettus.com/>. [Último acceso: 28 Agosto 2022].
- 43] | «National Instruments,» QUÉ ES LA RADIO DEFINIDA POR SOFTWARE USRP, 2022. [En línea]. Available: <https://www.ni.com/es-cr/shop/wireless-design-test/what-is-a-usrp-software-defined-radio.html>. [Último acceso: 28 Agosto 2022].
- 44] | Lab401, «Lab401,» 1, 2023. [En línea]. Available: <https://lab401.com/es/products/hackrf-one>. [Último acceso: 21 Noviembre 2023].
- 45] | J. SALAZAR, «Procesadores digitales de señal,» *Arquitecturas y criterios de selección*, [En línea]. Available: <https://www.um.es/documents/4874468/19345367/ssee-da-t04-01.pdf/20ed49bb-90d7-4d1b-a5d5-bc5cadc34fef>. [Último acceso: 28 Agosto 2022].
- 46] | «Gnu Radio,» 2022. [En línea]. Available: <https://www.gnuradio.org/>. [Último acceso: 27 Agosto 2022].
- 47] | P. Dominguez, «LABORATORIO DE COMUNICACIONES DIGITALES SDR,» Sevilla, Universidad de Sevilla, 2011, p. 31.

48] | «Análisis de los parámetros teóricos de las señales de RF con el uso de SDR
(HackRF One),» *Revista Iberica de Sistemas e Tecnologias de Informacao*, nº 25, pp.
1-15, 2020.

49] | A. N. E. AREVALO, «TRANSMISIÓN Y RECEPCIÓN DE IMAGEN CON
MODULACIÓN QPSK,» UNIVERSIDAD MILITAR NUEVA GRANADA,
BOGOTA, 2017.

50] | M. D.-S. Viader, DISEÑO E IMPLEMENTACIÓN DE UN FILTRO PASO
BANDA DE BANDA ESTRECHA CON TOPOLOGÍA INTERDIGITAL A
FRECUENCIAS UHF Y MICROONDAS, Bellaterra, 2007.

Anexos

Anexo 1: Sílabo de comunicaciones digitales 2023-2.



Universidad Estatal
Península de Santa Elena

Carrera de
Electrónica y Telecomunicaciones



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**



**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES
CARRERA DE TELECOMUNICACIONES**



**ADAPTACIÓN SÍLABO DE LA ASIGNATURA:
COMUNICACIONES DIGITALES**

DOCENTE

Ing. Fernando Chamba Macas. M.Sc.

PERÍODO ACADÉMICO

2023 - 2



UPSEec



@UPSE_ec



UPSE_ec



UPSE Santa Elena



UPSE noticias



flickr.com/upse

www.upse.edu.ec / djaramillo@upse.edu.ec

Cel: 0998106358

Vía La Libertad – Santa Elena



1.- Datos generales de la asignatura:

Asignatura:	Comunicaciones Digitales	Unidad de Organización Curricular:	Profesional
No. Créditos / Horas:	3/ 144	Semestre:	Sexto
Horas componente docencia:	48	Paralelo:	6/1
Horas componente A.P.E:	16	# semanas del semestre:	16
Horas componente T.A:	80	# horas semanales:	4
Pre-Requisito:	Fundamentos de Comunicaciones	Co-Requisito:	

2.- Datos específicos de la asignatura

<p>2.1 Contribución de la asignatura a la carrera y al perfil profesional:</p> <p>Una correcta aplicación y procesamiento de señales, con la finalidad de generar una sinergia con las materias de matemáticas, en especial con la ingeniería de Telecomunicaciones y la ingeniería en Electrónica</p>
<p>2.2 Caracterización de la asignatura</p> <p>Es una materia cuyo contenido permite conocer el fundamento teórico-práctico para el diseño y gestión de Sistemas de Transmisión Digital, transductores, canal de comunicación, ruido y Receptores o Demoduladores tanto para servicios de radiodifusión comercial, voz, datos y video.</p>
<p>2.3 Resultado de aprendizaje de la asignatura</p> <ul style="list-style-type: none"> • Aplicar los conocimientos adquiridos en los diferentes tipos de modulación. (Alta) • Aplicar habilidades en el manejo e interpretación de los datos tomados en los diferentes tipos de modulación. (Media) • Comprender el constante cambio tecnológico de los sistemas de comunicación y su compromiso de investigación continua. (Alta)



3.- Estructura del sílabo

N°	Unidades Temáticas	Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
			Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
1	Introducción	Describir las diversas etapas de procesamiento que hay en un sistema de comunicación digital y las ventajas de este tipo de sistema con respecto a uno analógico.	8 h	3	13 h	24 h
2	Transmisión en Banda Base	Explicar las características de las diversas variantes de la Transmisión en Banda Base de señales digitales y sus aplicaciones en los sistemas de comunicación digital.	8 h	3	13 h	24 h
3	Transmisión Pasa Banda. Técnicas de Modulación Digital	Analizar los procesos que se llevan a cabo sobre las señales digitales durante su transmisión a través de un sistema de comunicación digital de pasa banda y examinar las formas de determinar y mejorar el desempeño de estos sistemas ante el ruido.	8 h	2	11 h	21 h
4	Sistemas de Espectro Ensanchado	Analizar y comparar las diversas técnicas de modulación utilizadas en los sistemas de Espectro Ensanchado, examinar las formas de evaluar y mejorar el desempeño de estos sistemas ante el ruido y seleccionará la técnica de modulación más adecuada para cada necesidad de transmisión.	8 h	3	15 h	26 h
5	Transmisión de Datos	Describir el funcionamiento de las técnicas de transmisión de datos, comparar y explicar aplicaciones que tiene en los sistemas de comunicaciones.	8 h	3	13 h	24 h
6	Teoría de la Información y Codificación Fuente.	Describir el funcionamiento de la Teoría de la Información y Codificación Fuente, explicar las ventajas y desventajas de esta teoría y su aplicación en sistemas de comunicaciones.	8 h	2	15 h	25 h



4.- Sistema de contenidos (desagregación de las unidades temáticas)

Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
1	Introducción		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
1.1	Introducción a un Sistema de Comunicación	Conocer los sistemas de comunicación digital actual y como han ido evolucionando a través del tiempo.	2	-	2	4
1.2	Sistemas de comunicación Analógicos y Digitales	Conocer como está constituido en bloques de un Sistema de Comunicación de Datos.	2	-	2	4
1.3	Ventajas y Desventajas entre Sistemas de Comunicación Analógicos y Digitales	Conocer y diferenciar las ventajas y desventajas que tiene una transmisión digital con respecto a una transmisión analógica.	2	-	2	4
1.4	Conversión analógico Digital	Conocer el proceso que conlleva la conversión analógico a Digital.	2	-	3	5
	Talleres	Aplicar los métodos adecuados para diseñar un sistema de comunicación digital.	-	3	2	5
-	Prueba de conocimientos	Aplicar los métodos adecuados para diseñar un sistema de comunicación digital.	-	-	2	2
TOTAL:			8	3	13	24
Metodología:		Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Diapositivas, Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams				
Métodos, Técnicas e Instrumentos de Evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía Básica:		Sistemas de Comunicaciones Digitales y analógicos	Leon W. Couch	Pág. 02 – 30;86-93		
Bibliografía Complementaria:		Sistema de Comunicaciones Electrónicas	Wayne Tomasi	Pág. 667-668		
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
2	Transmisión en Banda Base		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
2.1	Formato de la banda base para la transmisión de datos.	Aprender en que consiste el formato de la banda base para la transmisión de datos.	2	-	2	4
2.2	Señales Discretas PAM.	Explicar las señales discretas PAM.	1	-	1	2



2.3	Espectro de potencia de las señales discretas PAM.	Explicar el espectro de potencia de las señales discreta PAM.	1		2	3
2.4	Desempeño frente al error de los códigos binarios.	Aprender el desempeño frente al error de los códigos binarios.	1		2	3
2.5	Criterio de Nyquist para la transmisión binaria en banda base sin distorsión	Aprender el criterio de Nyquist para la transmisión binaria en banda base sin distorsión.	1		-	1
2.6	Receptores óptimos para comunicaciones digitales.	Aprender sobre receptores óptimos para comunicaciones digitales.	2		-	2
-	Talleres	Aplicar los métodos técnicos para evaluar sistemas que utilizan Transmisión en Banda Base.	-	2	4	6
-	Prueba de conocimientos	Aplicar los métodos técnicos para evaluar sistemas que utilizan Transmisión en Banda Base.	-	1	2	3
Total:			8	3	13	24
Metodología:		Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Diapositivas, Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Sistemas de Comunicaciones Digitales y analógicos		Leon W. Couch	Pág. 128 - 230	
Bibliografía complementaria:		Sistemas de Comunicaciones Electrónicas		Wayne Tomasi	Pág. 746-748	
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
3	Transmisión Pasa Banda. Técnicas de modulación Digital.		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
3.1	Concepto de Ancho de Banda	Determinar los aspectos introductorios y teóricos relacionados al ancho de banda.	1	-	1	2
3.2	Tipos de Receptores.	Entender los tipos de receptores en un sistema de comunicación digital en pasa banda.	1	-	1	2
3.3	Sistemas Básicos de Modulación digital: ASK, FSK, Y PSK	Describir los diferentes sistemas de modulación.	1	-	1	2
3.4	Comparación de los sistemas de modulación binarios.	Describir los sistemas de modulación binarios.	1	-	1	2
3.5	Modulación Multinivel. M-PSK.	Describir la modulación multinivel M-PSK.	1		1	2
3.6	Modulación por corrimiento de fase en cuadratura(QPSK).	Determinar la modulación QPSK .	0		1	1



3.7	Modulación de amplitud en cuadratura QAM.	Describir la modulación QAM.	1		2	3
3.8	Modulación por corrimiento en fase M-ario(M-FSK)	Describir el corrimiento en fase Mario(M-FSK).	2		2	4
-	Talleres	Aplicar las diferentes técnicas de codificación de línea para las señales digitales	0	2	-	2
-	Prueba de conocimientos	Aplicar las diferentes técnicas de modulación digital.	0	-	1	1
Total:			8	2	11	21
Metodología:		Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Diapositivas, Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Sistemas de Comunicaciones Digitales y analógicos		Leon W. Couch	Pág. 230 - 301	
Bibliografía complementaria:		Sistemas de Comunicaciones Electronica		Wayne Tomasi	Pág. 467-524	
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
4	Sistema de Espectro Ensamchado.		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
4.1	Introducción a los sistemas de espectro ensanchado.	Describir en diagramas de bloques como está constituido un sistema de espectro ensanchado.	2	-	2	4
4.2	Sistemas de Espectro ensanchado: Secuencia Directa.	Describir la secuencia directa de un sistema de espectro ensanchado.	1	-	2	3
4.3	Salto de Frecuencia.	Describir la técnica de salto de frecuencia.	2	-	3	5
4.4	Multiplexación por división de frecuencia Ortogonal(OFDM).	Describir la técnicas de multiplexación por división de frecuencia ortogonal.	2	-	2	4
4.5	Sistema de acceso múltiple por división de código.	Describir el sistema de acceso múltiple por división de frecuencia .	1		2	3
-	Talleres	Aplicar las diferentes técnicas de modulación para transmitir varias señales digitales	-	2	2	4
-	Prueba de conocimientos	Aplicar las diferentes técnicas de modulación para transmitir varias señales digitales	-	1	2	3
Total:			8	3	15	26



Metodología:		Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Diapositivas, Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Sistemas de Comunicaciones Digitales y analógicos			Leon W. Couch	Pág. 367-379
Bibliografía complementaria:		Sistemas de Comunicaciones Electronica			Wayne Tomasi	Pág. 743-744
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
5	Transmisión de Datos		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
5.1	Velocidades de Transmisión.	Aprender sobre las velocidades de transmisión	1	-	1	2
5.2	El canal Digital.	Aprender los fundamentos técnicos como funciona el canal digital.	1	-	2	3
5.3	Clasificación de los canales .	Explicar sobre la clasificación de los canales.	1	-	2	3
5.4	Técnicas de Conmutación.	Explicar sobre las diferentes técnicas de conmutación.	1	-	2	3
5.5	El Modelo de Referencia ISO/OSI.	Explicar el modelo de referencia ISO/OSI.	1	-	2	3
5.6	Aspectos prácticos de la Modulación Digital.	Describir los aspectos prácticos de la modulación digital.	1	-	2	3
-	Talleres	Aplicar las diferentes técnicas de transmisión de datos.	-	2	2	4
-	Prueba de conocimientos	Aplicar las diferentes técnicas de transmisión de datos.	2	1	-	3
Total:			8	3	13	24
Metodología:		Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Diapositivas, Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Sistemas de Comunicaciones Digitales y analógicos			Leon W. Couch	Pág. 160-179
Bibliografía complementaria:		Transmisión de Datos.			José Briceño	Pág 1-34
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
g	Teoría de la Información y Codificación Fuente.		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
6.1	Capacidad de un Canal.	Estudiar las técnicas para calcular la capacidad de un canal de comunicaciones	1	-	2	3



6.2	Modelos de fuente de información y sus características.	Aplicar la transformada de Fourier inversa rápida IFFT para generar subportadoras	1	-	3	4
6.3	Medidas de Información: Información y Entropía.	Describir los parámetros reales de OFDM	1	-	3	4
6.4	Codificación de Fuente.	Describir el procesamiento de las señales OFDM	2	-	3	5
6.5	Aplicaciones	Describir las aplicaciones en los servicios de Tele.	1	-	3	4
-	Talleres	Aplicación de modelos y medidas de información	-	2	1	3
-	Prueba de conocimientos	Teoría de información y codificación	2	-	-	2
Total:			8	2	15	25
Metodología:		Métodos Expositivo-Ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Diapositivas, Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Sistemas de Comunicaciones Digitales y analógicos			Leon W. Couch	Pág. 16-32;367-372
Bibliografía complementaria:		Modulación Multiportadora OFDM			Leonardo Jimenez	Pág. 01-05



5.- Escenarios de aprendizajes

Escenario real		Escenario virtual
Áulico:	Salón de clases	Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams
Lugar de praxis:	-	

6.- Criterios normativos para la evaluación de la asignatura:

Tipos de Evaluación	Técnicas evaluativas	Ponderación	
		Primer Ciclo	Segundo Ciclo
Diagnóstica	Prueba de conocimientos previos	0	0
Formativa (Evaluación Continua)	ACD: Actuación de clases (5) , Lección Oral y Escritas (5); Prueba: resolución de ejercicios y problemas (15)	25%	25%
	APE: Talleres, prácticas	10%	10%
	AA: Trabajos, tareas (10), proyecto (15)	25%	25%
Sumativa	Examen de fin de ciclo	40%	40%
Recuperación			

7.- Bibliografía

Bibliografía Básica			
Autor	Nombre	Edición	Año
Leon W. Couch	Sistemas de Comunicaciones Digitales y Analógicos	7	2008
Bibliografía Complementaria			
Autor	Nombre	Edición	Año
Wayne Tomasi	Sistemas de Comunicaciones Electrónicas	4	2003
Artículo Científico			
Jaramillo D. Chuquimarca L. (2022). Estudio comprensivo de la Transformada de Fourier Discreta para el análisis de señales digitales. CTU, 75-84.			

8. Perfil del docente

Grado Académico	Tercer nivel	Ingeniero Electrónico
	Cuarto Nivel	Magister en Telecomunicaciones
Datos de Referencia	e-mail	djaramillo@upse.edu.ec
	# de teléfono	0998106358
Horario	Miércoles	13:00 – 15:00
	Viernes	15:30 – 17:30

9.- Políticas del curso



Universidad Estatal
Península de Santa Elena

Carrera de Electrónica y Telecomunicaciones

- ☑ El curso estará guiado por principios éticos: transparencia, respeto a la diversidad cultural, social y de capacidades físicas e intelectuales, enalteciéndose los valores humanos como la honestidad, la solidaridad, la responsabilidad y el humanismo.
- ☑ Se desarrollará un proceso enseñanza-aprendizaje activo, participativo, reflexivo, problemizador y cuestionador.
- ☑ Se utilizarán variados medios, enfatizándose en los audiovisuales y el empleo de la plataforma Moodle.
- ☑ Se procurará el desarrollo progresivo de habilidades intelectuales e investigativas.

Profesor responsable elaboración de sílabo	Director de Carrera	Revisado por:
Ing. Fernando Chamba M. M.Sc.	Ing. Ronald Rovira, Ph.D	
Fecha: 05-06-2023	Fecha:	Fecha:

Anexo 2: Sílabo de Procesamiento digital de señales 2023-2.

Universidad Estatal
Península de Santa Elena

Carrera de
Electrónica y Telecomunicaciones



**UNIVERSIDAD ESTATAL
PENÍNSULA DE SANTA ELENA**



**FACULTAD DE SISTEMAS Y
TELECOMUNICACIONES**

CARRERA DE TELECOMUNICACIONES



**ADAPTACIÓN SÍLABO DE LA ASIGNATURA:
PROCESAMIENTO DIGITAL DE SEÑALES**

DOCENTE

Ing. Daniel Armando Jaramillo Chamba M.Sc

PERÍODO ACADÉMICO

2023 -2



UPSEec



@UPSE_ec



UPSE_ec



UPSE Santa Elena



UPSE noticias



flickr.com/upse

www.upse.edu.ec / djaramillo@upse.edu.ec

Cel: 0998106358

Vía La Libertad - Santa Elena



Universidad Estatal
Península de Santa Elena

Carrera de Electrónica y Telecomunicaciones

1.- Datos generales de la asignatura:

Asignatura:	Procesamiento Digital de Señales	Unidad de Organización Curricular:	Básica
No. Créditos / Horas:	3/ 48	Semestre:	Sexto
Horas componente docencia:	48	Paralelo:	6/1
Horas aprendizaje práctico experimental	16	# semanas del semestre:	16
Horas componente T.A:	80	# horas semanales:	4
Pre-Requisito:	Análisis de señales y sistemas	Co-Requisito:	-

2.- Datos específicos de la asignatura

2.1 Contribución de la asignatura a la carrera y al perfil profesional:
Esta asignatura aporta al perfil del Ingeniero Electrónico la habilidad para Interpretar y diseñar sistemas digitales utilizados en diferentes procesos industriales, medición, instrumentación y sistemas de telecomunicaciones.
2.2 Caracterización de la asignatura
Se presenta la clasificación de las señales, frecuencia en señales en tiempo continuo y en tiempo discreto, conversión analógica digital, muestreo de señales análogas, teorema de muestreo, elementos básicos de un sistema de procesado digital de señales. Después se presenta las transformadas donde se trata en particular funciones de variable natural (funciones discretas en el tiempo), funciones discretas básicas, transformadas de las funciones discretas básicas, tablas de transformadas, ecuaciones diferenciales con diferencias, solución de ecuaciones de diferencias por el método de la transformada de Fourier, DFT, FFT, de coseno, Z y la transformada Z inversa. La parte final trata sobre el diseño de filtros IIR y FIR. En el último tema se presenta la aplicación, con un proyecto final implementando un sistema de adquisición de datos y tratamiento de la señal.
2.3 Resultado de aprendizaje de la asignatura
Implementar sistemas para el procesado de señales digitales, utilizando microprocesadores, DSP'S.



3.- Estructura del sílabo

N°	Unidades Temáticas	Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
			Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
1	Introducción al Procesamiento de Señales	Conocer los principales conceptos del Procesamiento Digital de Señales y los fundamentos matemáticos básicos para su aplicación, análisis y diseño, comprendiendo que es la base necesaria del curso	8 h	2 h	16 h	26 h
2	Modelamiento de Señales y Sistemas Discretos	Estudiar los Sistemas Lineales invariantes al tiempo y el uso de la Ecuación en Diferencia. Asimismo, repasar los conceptos de correlación y convolución discreta	8 h	3h	16 h	27 h
3	Transformada Discreta de Fourier	Analizar y simular por computadora la aplicación de la transformada discreta de Fourier (DFT) y la Transformada Discreta del Coseno. Ilustrar uno de los algoritmos más importantes del tratamiento de señales: la Transformada Rápida de Fourier (FFT)	12 h	4 h	16 h	32 h
4	Transformada Z. Aplicaciones	Presentar la Transformada Z como la transformada discreta de Laplace e ilustra su aplicación trascendental en los sistemas digitales	12 h	4 h	16 h	32 h
5	Diseño de Filtros Digitales Invariantes al Tiempo	Presentar los métodos para el diseño y aplicaciones de los filtros digitales no recursivos y recursivos, basadas en la aplicación de la Transformada Fourier.	8 h	3 h	16 h	27 h

4.- Sistema de contenidos (desagregación de las unidades temáticas)

Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas
			Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
1	Introducción al Procesamiento Digital de Señales					
1.1	Señales	Conocer las señales analógicas.	1	-	2	3



1.2	Clasificación de las señales.	Conocer la diferencia entre los distintos tipos de señales que existen en la naturaleza.	1	-	2	3
1.3	Muestreo de señales	Conocer proporción existente entre la potencia de la señal que se transmite y la potencia del ruido que la corrompe	1	-	2	3
1.4	Cuantificación de señales	Caracterizar y analizar las señales	1	-	2	3
1.5	Señales discretas en el tiempo	Reconocer las funciones discretas	1	-	2	3
1.6	Clasificación de las señales discretas	Revisar las funciones más comunes en el PDS	1	-	2	3
1.7	Manipulación de señales discretas	Estudiar las diferentes formas de manipular la señales	2	-	4	6
1.8	Talleres	Aplicación experimental de los temas tratados en la unidad	-	2	-	2
TOTAL:			8	2	16	26
Metodología:		Método Expositivo-ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Plataforma Moodle (Aula Virtual), Diapositivas, Zoom, Microsoft Teams.				
Métodos, Técnicas e Instrumentos de Evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía Básica:		Procesamiento Digital de Señales. 4ta Ed.	Manolakis	Pág. 01-43		
Bibliografía Complementaria:		Procesamiento Digital de Señales. 4ta Ed.	Manolakis	Pág. 01-43		
Unidad Temática	Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje			Total de Horas	
2	Modelamiento de Señales y Sistemas Discretos		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
2.1	Sistemas Lineales Discretos e invariantes en el tiempo	Analizar y caracterizar los sistemas lineales e invariantes en el tiempo y sus propiedades	1	-	3	4
2.2	Sistemas descritos mediante ecuaciones en diferencias	Validar el desempeño de los modelos en el dominio discreto	1	-	3	4
2.3	Cálculo de la Convolución Numérica y Gráfica	Describir numéricamente el resultado de un experimento aleatorio	2	-	3	5
2.4	Correlación entre Señales Discretas	Explicar y modelar las principales señales discretas utilizadas en los sistemas de comunicación digital	2	-	3	5
-	Talleres	Aplicar los métodos adecuados para analizar los sistemas de comunicación digital	-	3	2	5
-	Prueba de conocimientos	Aplicar los métodos adecuados para analizar los sistemas de comunicación digital	2	-	2	4
Total:			8	3	16	27



Universidad Estatal
Península de Santa Elena

Carrera de Electrónica y Telecomunicaciones

Metodología:		Método Expositivo-ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Plataforma Moodle (Aula Virtual), Diapositivas, Zoom, Microsoft Teams.				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Procesamiento Digital de Señales. 4ta Ed.		Manolakis	Pág. 43 - 151	
Bibliografía complementaria:		Procesamiento de la señal en el tiempo discreto 2da Ed		Oppenheim	Pág. 140-210	
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje		Total de Horas	
3	Transformada Discreta de Fourier		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
3.1	Propiedades de la transformada Discreta de Fourier (DFT)	Analizar señales empleando de la transformada de Fourier en tiempo discreto y la transformada Discreta de Fourier	2	-	3	5
3.2	Fundamentos de la Transformada Rápida de Fourier o FFT. Algoritmos	Analizar Algoritmos su decimación de tiempo y decimación de frecuencia empleando la Transformada Rápida de Fourier	2	-	3	5
3.3	Transformada de Fourier Inversa (IDFT)	Analizar el cambio de dominio de una señal en el dominio de frecuencia al dominio del tiempo.	2	-	3	5
3.4	Transformada Discreta del Coseno (DCT)	Concentrar la mayor parte de la información en pocos coeficientes transformados	4	-	3	7
-	Talleres	Aplicar los métodos adecuados para analizar la transformada de Fourier	-	4	2	6
-	Prueba de conocimientos	Aplicar los métodos adecuados para analizar la transformada Fourier	2	-	2	4
Total:			12	4	16	32
Metodología:		Método Expositivo-ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Plataforma Moodle (Aula Virtual), Diapositivas, Zoom, Microsoft Teams.				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Procesamiento Digital de Señales. 4ta Ed.		Manolakis	Pág. 461-523	
Bibliografía complementaria:		Procesamiento de la señal en el tiempo discreto 2da Ed		Oppenheim	Pág. 693-748	
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje		Total de Horas	
4	Transformada Z directa e inversa. Propiedades		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
4.1	Definición de transformada Z	Esbozar en el plano Z la transformada z directa.	2	-	3	5



Universidad Estatal
Península de Santa Elena

Carrera de Electrónica y Telecomunicaciones

4.2	Función de transferencia	Analizar la función de transferencia de la transformada Z	2	-	3	5
4.3	Transformada Z inversa	Manejar la transformada Z inversa	2	-	3	5
4.4	Análisis en el dominio Z	Analizar la respuesta de los sistemas en el dominio Z	4	-	3	7
-	Talleres	Aplicar los métodos adecuados para analizar la transformada de Fourier	-	4	2	4
-	Prueba de conocimientos	Aplicar los métodos adecuados para analizar la transformada Fourier	2	-	2	4
Total:			12	4	16	32
Metodología:		Método Expositivo-ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Plataforma Moodle (Aula Virtual), Diapositivas, Zoom, Microsoft Teams.				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				
Bibliografía básica:		Procesamiento Digital de Señales. 4ta Ed.		Manolakis	Pág. 151 - 229	
Bibliografía complementaria:		Procesamiento de la señal en el tiempo discreto 2da E		Oppenheim	Pág. 94 - 126	
Unidad Temática		Resultados de aprendizaje (Unidades Temáticas)	Componentes de gestión del aprendizaje		Total de Horas	
5	Diseño de Filtros Digitales		Docencia Asistida por el docente y colaborativo	Aplicación y experimentación de los aprendizajes	Trabajo autónomo	
5.1	Concepto y características de los filtros digitales	Describir los métodos básicos para el diseño de filtros FIR	2	-	3	5
5.2	Filtros FIR	Diseñar filtros FIR mediante el método de la ventana	2	-	3	5
5.3	Respuesta en el Tiempo y Frecuencia de los Filtros FIR	Caracterizar y describir matemáticamente filtros FIR	1	-	3	4
5.4	Filtros IIR	Analizar la Respuesta en el Tiempo y Frecuencia y su Operación de Filtrado con Filtros IIR	1	-	3	4
-	Talleres	Aplicar los métodos adecuados para el diseño y aplicaciones de los filtros digitales no recursivos y recursivos	-	3	2	5
-	Prueba de conocimientos	Aplicar los métodos adecuados para el diseño y aplicaciones de los filtros digitales no recursivos y recursivos	2	-	2	4
Total:			8	3	16	27
Metodología:		Método Expositivo-ilustrativo; Resolución de ejercicios y problemas; Aprendizaje cooperativo.				
Recursos Didácticos:		Plataforma Moodle (Aula Virtual), Diapositivas, Zoom, Microsoft Teams.				
Métodos, técnicas e instrumentos de evaluación:		Preguntas Orales y Escritas, Trabajos, Lecciones, Resolución de ejercicios y problemas				



Universidad Estatal
Península de Santa Elena

Carrera de Electrónica y Telecomunicaciones

Bibliografía básica:	Procesamiento Digital de Señales. 4ta Ed.	Manolakis	Pág. 669-767
Bibliografía complementaria:	Procesamiento de la señal en el tiempo discreto 2da E	Oppenheim	Pág. 439-510



5.- Escenarios de aprendizajes

	Escenario real	Escenario virtual
Áulico:	B2-201	Plataforma Moodle (Aula Virtual), Zoom, Microsoft Teams
Lugar de praxis:	-	

6.- Criterios normativos para la evaluación de la asignatura:

Tipos de Evaluación	Técnicas evaluativas	Ponderación		
		Primer Ciclo	Segundo Ciclo	
Diagnóstica	Prueba de conocimientos previos	0	0	
Formativa (Evaluación Continua)	Docencia:	Preguntas Orales y Escritas; lecciones, resolución de ejercicios y problemas	25%	25%
	Aplicación y Experimentación:	Talleres, simulaciones	10%	10%
	Trabajo Autónomo:	Tareas, Proyecto	25%	25%
Sumativa	Examen de fin de ciclo	40%	40%	
Recuperación	Se considerará el promedio entre la calificación del examen de recuperación con una valoración del 100% y la calificación más alta de los ciclos I y II del periodo académico.			

7.- Bibliografía

Bibliografía Básica			
Autor	Nombre	Edición	Año
Proakis John G.	Procesamiento Digital de Señales	4	2014
Bibliografía Complementaria			
Autor	Nombre	Edición	Año
Oppenheim Alan V.	Tratamiento Digital de señales en el tiempo Discreto	3	2011
Artículo Científico			
Jaramillo D. Chuquimarca L. (2022). Estudio comprensivo de la Transformada de Fourier Discreta para el análisis de señales digitales. CTU, 75-84.			

8. Perfil del docente

Grado Académico	Tercer nivel	Ingeniero Electrónico
	Cuarto Nivel	Magister en Telecomunicaciones
Datos de Referencia	e-mail	djaramillo@upse.edu.ec
	# de teléfono	0998106358
Horario:		

9.- Políticas del curso



Universidad Estatal
Península de Santa Elena

Carrera de Electrónica y Telecomunicaciones

- ☑ El curso estará guiado por principios éticos: transparencia, respeto a la diversidad cultural, social y de capacidades físicas e intelectuales, enalteciéndose los valores humanos como la honestidad, la solidaridad, la responsabilidad y el humanismo.
- ☑ Se desarrollará un proceso enseñanza-aprendizaje activo, participativo, reflexivo, problemizador y cuestionador.
- ☑ Se utilizarán variados medios, enfatizándose en los audiovisuales y el empleo de la plataforma Moodle.
- ☑ Se procurará el desarrollo progresivo de habilidades intelectuales e investigativas.

Profesor responsable elaboración de sílabo	Director de Carrera	Revisado por:
		
Ing. Daniel Jaramillo Ch.	Ing. Ronald Rovira, Ph.D.	
Fecha: 05-09-2023	Fecha:	Fecha:

Anexo 3: Factura de HackRF One.**Detalles finales del pedido #114-1032168-5403443**

[Imprima esta página para sus registros.](#)

Pedido realizado: 1 de agosto de 2022
 Pedido amazon.com número: 114-1032168-5403443
 Total del pedido: \$347.70

Enviado el 1 de agosto de 2022

Productos comprados:	Precio
1 de: <i>Paquete de radio definida por software HackRF One (SDR) y antena ANT500</i>	\$324.95
Vendido por: Nooelec (Perfil del vendedor) ¿Dudas sobre el producto? Pregunta al vendedor	
Suministrado por: Otro	
Estado: Nuevo	

Dirección de envío:
 CESAR JACHO NAVIA # EC29536
 2301 NW 107TH AVE STE 102
 DORAL, FL 33172-2242
 Estados Unidos

Velocidad de envío:
 GRATIS Entrega Prime

Información de pago

Método de pago:	Productos: \$324.95
Visa que termina en 2898	Envío: \$0.00

Dirección de facturación:	Total antes de impuestos: \$324.95
CESAR JACHO NAVIA # EC29536	Impuestos: \$22.75
2301 NW 107TH AVE STE 102	---
DORAL, FL 33172-2242	Total (I.V.A. Incluido): \$347.70
Estados Unidos	

Transacciones con la tarjeta de crédito Visa que termina en 2898: 1 de agosto de 2022: \$347.70

Para ver el estado de tu pedido, vuelve a [Resumen del pedido](#).

Anexo 4: Factura de computadora completa.



ESCOBAR OLAYA DAVID GONZALO
 Dirección Matriz: Urb. Castilla mz 29 sl 5
 Teléfono 1: 045025276 Teléfono 2: 0939721182
 Correo: ventas@dhecomputers.com

Obligado a Llevar Contabilidad: NO
 Contribuyente Régimen RIMPE

RUC.: 0955926241001
 FACTURA N. 001-100-00000040
 FECHA AUTORIZACION : 09/11/2023 16:08:00
 AMBIENTE: PRODUCCION
 EMISION: NORMAL

NUMERO DE AUTORIZACION:

0911202301095592624100120011000000000401234567810

CLAVE DE ACCESO:



Nombres: Ruby Monik Sáenz Valero

Dirección: Cdla virgen del Carmen cll 15 av 33

RUC: 0605493923

Fecha Emision: 09/11/2023

Vence: 09/11/2023

Telefono: 0-0996631655-0

Código	Descripción	Med.	Cantidad	Precio Unitario	Desc.	Total
P000000022	CPU Intel i5 8ava /8Gb	Unidad	1,00	401,785700	0,00	401,79
P000000004	MONITOR DE 19 PULGADAS WS	Unidad	1,00	40,178571	0,00	40,18
P000000010	COMBO TECLADO Y MOUSE	Unidad	1,00	4,464300	0,00	4,46
Forma Pago			Subtotal:		446,43	
Sin Utilización Del Sistema Financiero 500,00			Descuento:		0,00	
INFORMACIÓN ADICIONAL			Subtotal Neto:		446,43	
vendedor : Vendedor			Subtotal Con Impuestos:		446,43	
			Subtotal Impuestos 0%:		0,00	
			Subtotal No Objeto IVA:		0,00	
			Subtotal Exento IVA :		0,00	
			ICE:		0,00	
			IVA 12%:		53,57	
			Propina:		0,00	
			VALOR TOTAL		500,00	

Anexo 5 :Código de modulaciones QAM en MATLAB

```

function varargout = MODULACIONES_QAM(varargin)
% MODULACIONES_QAM MATLAB code for MODULACIONES_QAM.fig
%   MODULACIONES_QAM, by itself, creates a new MODULACIONES_QAM or raises
the existing
%   singleton*.
%
%   H = MODULACIONES_QAM returns the handle to a new MODULACIONES_QAM
or the handle to
%   the existing singleton*.
%
%   MODULACIONES_QAM('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in MODULACIONES_QAM.M with the given input
arguments.
%
%   MODULACIONES_QAM('Property','Value',...) creates a new
MODULACIONES_QAM or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before MODULACIONES_QAM_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to MODULACIONES_QAM_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MODULACIONES_QAM

% Last Modified by GUIDE v2.5 30-Nov-2023 13:17:28

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @MODULACIONES_QAM_OpeningFcn, ...
    'gui_OutputFcn', @MODULACIONES_QAM_OutputFcn, ...
    'gui_LayoutFcn', [], ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

```

```

% --- Executes just before MODULACIONES_QAM is made visible.
function MODULACIONES_QAM_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MODULACIONES_QAM (see VARARGIN)

% Choose default command line output for MODULACIONES_QAM
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

axes(handles.logo);%Mostrar grafica logo1
path = 'logo1.jpg';
img = imread(path);
imshow(img);
axis off;

axes(handles.constelacion);% Muestra los cuadros donde aparecen cantelación, awgn y ber.
axis on;
axes(handles.AWGN);
axis on;
axes(handles.BEAMFORMING);
axis on;

set(handles.banda,'visible','off');
set(handles.canales,'visible','off');
set(handles.datarate,'visible','off');
set(handles.uibuttongroup4,'visible','off');
set(handles.uibuttongroup5,'visible','off');
set(handles.uibuttongroup6,'visible','off');
set(handles.uibuttongroup7,'visible','off');

global M_QAM banda datar
M_QAM=64
banda=24
datar=450

% --- Outputs from this function are returned to the command line.
function varargout = MODULACIONES_QAM_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Get default command line output from handles structure

```

```
varargout{1} = handles.output;
```

```
% --- Executes on button press in Iniciar.
```

```
function Iniciar_Callback(hObject, eventdata, handles)% Boton iniciar
```

```
% hObject handle to Iniciar (see GCBO)
```

```
% eventdata reserved - to be defined in a future version of MATLAB
```

```
% handles structure with handles and user data (see GUIDATA)
```

```
global M_QAM banda datar
```

```
    cla(handles.constelacion,'reset') % Limpiar grafica
```

```
    cla(handles.AWGN,'reset') % Limpiar grafica
```

```
    cla(handles.BEAMFORMING,'reset') % Limpiar grafica
```

```
% Limpia los cuadros donde aparecen cantelación, awgn y ber.
```

```
numSymPerFrame = get(handles.trama,'string') %#ok<*NOPRT,*NASGU>%obtenemos el
número de bits x simbolos de trama
```

```
disp(numSymPerFrame)%imprimimos en Command Window el numero
```

```
numSymPerFrame= str2num(numSymPerFrame) %#ok<*ST2NM>
```

```
assignin('base','numSymPerFrame',numSymPerFrame);%Guardamos esta variable en el
Workspace principal
```

```
Th = get(handles.canales,'string') %#ok<*NOPRT,*NASGU>%obtenemos el numero canales
disp(Th)%imprimimos en Command Window el numero
```

```
Th= str2num(Th) %#ok<*ST2NM>
```

```
assignin('base','Th',Th);%Guardamos esta variable en el Workspace principal
```

```
if (numSymPerFrame > 0) && (Th > 0)
```

```
if (M_QAM==8)
```

```
    M = 8; % Modulation order
```

```
    x = (0:7); % Integer input
```

```
    symbin = qammod(x,M,'bin'); % 16-QAM output (natural-coded binary)
```

```
    %symgray = qammod(x,M,'gray'); % 16-QAM output (Gray-coded)
```

```
    %scatterplot(symgray,1,0,'b*');
```

```
    scatterplot(symbin,1,0,'b.');
```

```
    for k = 1:M
```

```
        if k>0 && k<11
```

```
            text(real(symbin(k))-0.25, imag(symbin(k))-0.7,...
```

```
                num2str(x(k),'FontSize',7.5);
```

```
        elseif k>10 && k<101
```

```
            text(real(symbin(k))-0.5, imag(symbin(k))-0.7,...
```

```
                num2str(x(k),'FontSize',7.5);
```

```
        elseif k>100
```

```
            text(real(symbin(k))-0.8, imag(symbin(k))-0.7,...
```

```
                num2str(x(k),'FontSize',7.5);
```

```
    end
```

```
end
```

```
axis([-4 4 -4 4])
```

```

saveas(gcf,'Constelacion.png')% Guardar figura como imagen
hold off;
title('8-QAM ')% Título para el diagrama de constelacion

axes(handles.constelacion);%Mostrar grafica diagrama de dispersion guardado
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;
%-----
k = log2(M); % Number of bits per symbol
n = 30000; % Number of bits to process
sps = 1; % Number of samples per symbol (oversampling factor)

%datos binarios aleatorios
rng default;
dataIn = randi([0 1],n,1); % Generate vector of binary data

%convierte los valores binarios a enteros para usar la funcion QAM (solo usa valores
enteros)
dataInMatrix = reshape(dataIn,length(dataIn)/k,k);
dataSymbolsIn = bi2de(dataInMatrix);

%modulacion con la funcion qammod (binaryo y gray)
dataMod = qammod(dataSymbolsIn,M,'bin'); % Binary coding with phase offset of zero
dataModG = qammod(dataSymbolsIn,M); % Gray coding with phase offset of zero

% la señal ya modulada se la pasa por ruido gaussiano blanco
EbNo = 10;
snr = EbNo+10*log10(k)-10*log10(sps);

%asignacion de simbolos binarios y codificacion gray
receivedSignal = awgn(dataMod,snr,'measured');
receivedSignalG = awgn(dataModG,snr,'measured');

%Crear diagrama de constelación
%Utilice la función de diagrama de dispersión para mostrar los componentes en fase y
en
%cuadratura de la señal modulada, dataMod y la señal ruidosa recibida después del
canal.
%Los efectos de AWGN están presentes en el diagrama de constelaciones.
sPlotFig = scatterplot(receivedSignal,1,0,'g. ');
hold on
scatterplot(dataMod,1,0,'k.',sPlotFig)
saveas(gcf,'AWGN.png')% Guardar figura como imagen
hold off;
title('8-QAM ')% Título para el diagrama de constelacion

axes(handles.AWGN);%Mostrar grafica diagrama de dispersion guardado
path = 'AWGN.png';

```

```

img = imread(path);
imshow(img);
axis off;
%-----
axes(handles.BEAMFORMING);
k = log2(M);      % Bits per symbol
EbNoVec = (2:8);  % Eb/No values (dB)
numSymPerFrame = 100; % Number of QAM symbols per frame
berEst = zeros(size(EbNoVec));
for n = 1:length(EbNoVec)
    % Convert Eb/No to SNR
    snrdB = EbNoVec(n) + 10*log10(k);
    % Reset the error and bit counters
    numErrs = 0;
    numBits = 0;

    while numErrs < 200 && numBits < 1e7
        % Generate binary data and convert to symbols
        dataIn = randi([0 1],numSymPerFrame,k);
        dataSym = bi2de(dataIn);

        % QAM modulate using 'Gray' symbol mapping
        txSig = qammod(dataSym,M);

        % Pass through AWGN channel
        rxSig = awgn(txSig,snrdB,'measured');

        % Demodulate the noisy signal
        rxSym = qamdemod(rxSig,M);
        % Convert received symbols to bits
        dataOut = de2bi(rxSym,k);

        % Calculate the number of bit errors
        nErrors = biterr(dataIn,dataOut);

        % Increment the error and bit counters
        numErrs = numErrs + nErrors;
        numBits = numBits + numSymPerFrame*k;
    end

    % Estimate the BER
    berEst(n) = numErrs/numBits;
end
berTheory = berawgn(EbNoVec,'qam',M);
semilogy(EbNoVec,berEst, '*')
hold on
semilogy(EbNoVec,berTheory)
grid
legend('Estimated BER','Theoretical BER')
xlabel('Eb/No (dB)')

```

```

        ylabel('Bit Error Rate')

end

if (M_QAM==16)
    M = 16; % Modulation order
    x = (0:15); % Integer input
    symbin = qammod(x,M,'bin'); % 16-QAM output (natural-coded binary)
    %symgray = qammod(x,M,'gray'); % 16-QAM output (Gray-coded)
    %scatterplot(symgray,1,0,'b*');
    scatterplot(symbin,1,0,'b. ');
    for k = 1:M
        if k>0 && k<11
            text(real(symbin(k))-0.25, imag(symbin(k))-0.7,...
                num2str(x(k)), 'FontSize',7.5);
        elseif k>10 && k<101
            text(real(symbin(k))-0.5, imag(symbin(k))-0.7,...
                num2str(x(k)), 'FontSize',7.5);
        elseif k>100
            text(real(symbin(k))-0.8, imag(symbin(k))-0.7,...
                num2str(x(k)), 'FontSize',7.5);
        end
    end
    axis([-4 4 -4 4])
    saveas(gcf,'Constelacion.png')% Guardar figura como imagen
    hold off;
    title('8-QAM ')% Título para el diagrama de constelacion

    axes(handles.constelacion);%Mostrar grafica diagrama de dispersion guardado
    path = 'Constelacion.png';
    img = imread(path);
    imshow(img);
    axis off;

%-----
    k = log2(M); % Number of bits per symbol
    n = 30000; % Number of bits to process
    sps = 1; % Number of samples per symbol (oversampling factor)

    %datos binarios aleatorios
    rng default;
    dataIn = randi([0 1],n,1); % Generate vector of binary data

    %convierte los valores binarios a enteros para usar la funcion QAM (solo usa valores
    enteros)
    dataInMatrix = reshape(dataIn,length(dataIn)/k,k);
    dataSymbolsIn = bi2de(dataInMatrix);

    %modulacion con la funcion qammod (binaryo y gray)
    dataMod = qammod(dataSymbolsIn,M,'bin'); % Binary coding with phase offset of zero
    dataModG = qammod(dataSymbolsIn,M); % Gray coding with phase offset of zero

```

```

% la señal ya modulada se la pasa por ruido gaussiano blanco
EbNo = 10;
snr = EbNo+10*log10(k)-10*log10(sps);

% asignacion de simbolos binarios y codificacion gray
receivedSignal = awgn(dataMod,snr,'measured');
receivedSignalG = awgn(dataModG,snr,'measured');

% Crear diagrama de constelación
% Utilice la función de diagrama de dispersión para mostrar los componentes en fase y
en
% cuadratura de la señal modulada, dataMod y la señal ruidosa recibida después del
canal.
% Los efectos de AWGN están presentes en el diagrama de constelaciones.
sPlotFig = scatterplot(receivedSignal,1,0,'g. ');
hold on
scatterplot(dataMod,1,0,'k.',sPlotFig)
saveas(gcf,'AWGN.png')% Guardar figura como imagen
hold off;
title('8-QAM ' )% Título para el diagrama de constelacion

axes(handles.AWGN);% Mostrar grafica diagrama de dispersion guardado
path = 'AWGN.png';
img = imread(path);
imshow(img);
axis off;

%-----
axes(handles.BEAMFORMING);
M = 16;           % Modulation order
k = log2(M);     % Bits per symbol
EbNoVec = (2:9)'; % Eb/No values (dB)
numSymPerFrame = 50; % Number of QAM symbols per frame
berEst = zeros(size(EbNoVec));
for n = 1:length(EbNoVec)
    % Convert Eb/No to SNR
    snrdB = EbNoVec(n) + 10*log10(k);
    % Reset the error and bit counters
    numErrs = 0;
    numBits = 0;

    while numErrs < 200 && numBits < 1e7
        % Generate binary data and convert to symbols
        dataIn = randi([0 1],numSymPerFrame,k);
        dataSym = bi2de(dataIn);

        % QAM modulate using 'Gray' symbol mapping
        txSig = qammod(dataSym,M);

        % Pass through AWGN channel

```

```

rxSig = awgn(txSig,snrdB,'measured');

% Demodulate the noisy signal
rxSym = qamdemod(rxSig,M);
% Convert received symbols to bits
dataOut = de2bi(rxSym,k);

% Calculate the number of bit errors
nErrors = biterr(dataIn,dataOut);

% Increment the error and bit counters
numErrs = numErrs + nErrors;
numBits = numBits + numSymPerFrame*k;
end

% Estimate the BER
berEst(n) = numErrs/numBits;
end
berTheory = berawgn(EbNoVec,'qam',M);
semilogy(EbNoVec,berEst, '*')
hold on
semilogy(EbNoVec,berTheory)
grid
legend('Estimated BER','Theoretical BER')
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')
end

if (M_QAM==32)
M = 32; % Modulation order
x = (0:31); % Integer input
symbin = qammod(x,M,'bin'); % 16-QAM output (natural-coded binary)
% symgray = qammod(x,M,'gray'); % 16-QAM output (Gray-coded)
% scatterplot(symgray,1,0,'b*');
scatterplot(symbin,1,0,'b. ');
for k = 1:M
if k>0 && k<11
text(real(symbin(k))-0.25, imag(symbin(k))-0.7,...
num2str(x(k)), 'FontSize',7.5);
elseif k>10 && k<101
text(real(symbin(k))-0.5, imag(symbin(k))-0.7,...
num2str(x(k)), 'FontSize',7.5);
elseif k>100
text(real(symbin(k))-0.8, imag(symbin(k))-0.7,...
num2str(x(k)), 'FontSize',7.5);
end
end
axis([-6 6 -6 6])
saveas(gcf,'Constelacion.png')% Guardar figura como imagen
hold off;

```

```

title('8-QAM ')% Título para el diagrama de constelacion

axes(handles.constelacion);%Mostrar grafica diagrama de dispersion guardado
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;
%-----
k = log2(M); % Number of bits per symbol
n = 30000; % Number of bits to process
sps = 1; % Number of samples per symbol (oversampling factor)

%datos binarios aleatorios
rng default;
dataIn = randi([0 1],n,1); % Generate vector of binary data

%convierte los valores binarios a enteros para usar la funcion QAM (solo usa valores
enteros)
dataInMatrix = reshape(dataIn,length(dataIn)/k,k);
dataSymbolsIn = bi2de(dataInMatrix);

%modulacion con la funcion qammod (binaryo y gray)
dataMod = qammod(dataSymbolsIn,M,'bin'); % Binary coding with phase offset of zero
dataModG = qammod(dataSymbolsIn,M); % Gray coding with phase offset of zero

% la señal ya modulada se la pasa por ruido gaussiano blanco
EbNo = 10;
snr = EbNo+10*log10(k)-10*log10(sps);

%asignacion de simbolos binarios y codificacion gray
receivedSignal = awgn(dataMod,snr,'measured');
receivedSignalG = awgn(dataModG,snr,'measured');

%Crear diagrama de constelación
%Utilice la función de diagrama de dispersión para mostrar los componentes en fase y
en
%cuadratura de la señal modulada, dataMod y la señal ruidosarecibida después del
canal.
%Los efectos de AWGN están presentes en el diagrama de constelaciones.
sPlotFig = scatterplot(receivedSignal,1,0,'g. ');
hold on
scatterplot(dataMod,1,0,'k.',sPlotFig)
saveas(gcf,'AWGN.png')% Guardar figura como imagen
hold off;
title('8-QAM ')% Título para el diagrama de constelacion

axes(handles.AWGN);%Mostrar grafica diagrama de dispersion guardado
path = 'AWGN.png';
img = imread(path);
imshow(img);

```

```

axis off;
%-----
axes(handles.BEAMFORMING);
M = 32;           % Modulation order
k = log2(M);     % Bits per symbol
EbNoVec = (2:10)'; % Eb/No values (dB)
numSymPerFrame = 90; % Number of QAM symbols per frame
berEst = zeros(size(EbNoVec));
for n = 1:length(EbNoVec)
    % Convert Eb/No to SNR
    snrdB = EbNoVec(n) + 10*log10(k);
    % Reset the error and bit counters
    numErrs = 0;
    numBits = 0;

    while numErrs < 200 && numBits < 1e7
        % Generate binary data and convert to symbols
        dataIn = randi([0 1],numSymPerFrame,k);
        dataSym = bi2de(dataIn);

        % QAM modulate using 'Gray' symbol mapping
        txSig = qammod(dataSym,M);

        % Pass through AWGN channel
        rxSig = awgn(txSig,snrdB,'measured');

        % Demodulate the noisy signal
        rxSym = qamdemod(rxSig,M);
        % Convert received symbols to bits
        dataOut = de2bi(rxSym,k);

        % Calculate the number of bit errors
        nErrors = biterr(dataIn,dataOut);

        % Increment the error and bit counters
        numErrs = numErrs + nErrors;
        numBits = numBits + numSymPerFrame*k;
    end

    % Estimate the BER
    berEst(n) = numErrs/numBits;
end
berTheory = berawgn(EbNoVec,'qam',M);
semilogy(EbNoVec,berEst, '*')
hold on
semilogy(EbNoVec,berTheory)
grid
legend('Estimated BER','Theoretical BER')
xlabel('Eb/No (dB)')
ylabel('Bit Error Rate')

```

end

```
if (M_QAM==64 || M_QAM==128 || M_QAM==256)
```

```
if (M_QAM==64) && (banda==24 || banda==5) && (datar==450 ||
datar==600)% _____
```

```
%-----
%-----64-QAM-----
%-----
%-----
```

```
    % hold off;
```

```
% *****
% *****
```

```
%           Modulación de amplitud en cuadratura M-QAM
```

```
% *****
% *****
```

```
% Modular una secuencia de bits utilizando 64-QAM. Pase la señal a través de un canal
ruidoso.
```

```
% Muestre el diagrama de constelación resultante.
```

```
% Establezca el orden de modulación y determine el número de bits por símbolo.
```

```
M = 64;
```

```
k = log2 (M);
```

```
% numSymPerFrame = 10000; % Número de símbolos QAM por trama
```

```
% Cree una secuencia de datos binarios aleatorios. Cuando se utilizan entradas binarias, el
número de filas
```

```
% en la entrada debe ser un múltiplo entero del número de bits por símbolo.
```

```
data = randi([0 1],numSymPerFrame*k,1);% 1000
```

```
% Modular la señal utilizando entradas de bits y configurarla para que tenga una potencia
media unitaria.
```

```
txSig = qammod(data,M,'InputType','bit','UnitAveragePower',true);
```

```
% Pase la señal a través de un canal ruidoso AWGN.
```

```
% Pase la señal a través de un canal AWGN que tenga una relación señal-ruido (SNR) de 25
dB.
```

```
snrdB = 25;
```

```
rxSig = awgn (txSig, snrdB,'measured');
```

```
% _____
```

```
%           TRAZO DE DIAGRAMA DE CONSTELACION
```

```
% _____
```

```
% Traza el diagrama de la constelación de señales moduladas usando la función scatterplot.
```

```
% axes(handles.constelacion)
```

```

hold on;
scatterplot(txSig);
saveas(gcf,'Constelacion.png')% Guardar figura como imagen
hold off;
title('64-QAM ')% Título para el diagrama de constelacion

axes(handles.constelacion);%Mostrar grafica diagrama de dispersion guardado
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;

% Traza el diagrama de la constelación con ruido de canal AWGN.
% cd = comm.ConstellationDiagram('ShowReferenceConstellation',false);
% step(cd,rxSig)%Ejecutar algoritmo de objetos del sistema
% Traza el diagrama de la constelación con ruido de canal AWGN. con fondo
% blanco y constelaciones en azul
scatterplot(rxSig); %#ok<*NASGU>
saveas(gcf,'AWGN.png')% Guardar figura como imagen
hold on
%scatterplot([],[],'r*','h)
grid
hold off
axes(handles.AWGN);%Mostrar grafica diagrama de dispersion guardado
path = 'AWGN.png';
img = imread(path);
imshow(img);
axis off;

% *****
% *****
%          BER  Beamforming usando modulación M-QAM
% *****
% *****
%Beamforming usando modulación M-QAM en el canal de desvanecimiento Rayleigh
N = 9^6; % number of bits
M=64; %M-QAM [M=4,16,64,128...]
k = sqrt(1/((2/3)*(M-1))); % normalizing factor
m = [1:sqrt(M)/2]; %#ok<*NBRAK>
alphaMqam = [-(2*m-1) 2*m-1]; % symbols for M-QAM modulation
Eb_N0_dB = [0:25]; % multiple Eb/N0 values
ipHat = zeros(1,N); %#ok<*PREALL> % initialzng
% Th=3%#ok<*NOPRT> %input('Enter the number of Channel Taps :');%take input from
user(enter 5 )
for ii = 1:length(Eb_N0_dB)
    ip = randsrc(1,N,alphaMqam) + j*randsrc(1,N,alphaMqam);%generate input symbols
    s = k*ip; % normalization of energy to 1
    n = 1/sqrt(2)*[randn(1,N) + j*randn(1,N)]; %#ok<*IJCL> % white gaussian noise, 0dB
variance

```

```

h = 1/sqrt(2)*[randn(Th,N) + j*randn(Th,N)]; % Rayleigh fading channel
signal = (1/sqrt(Th))*kron(ones(Th,1),s); %signal
% Channel and noise addition
phase = exp(j*angle(h)); %calculate the phase of channel
phase_conjugate = conj(phase); %conjugate of phase
bf_ch = h.*phase_conjugate; %beamforming channel
y1 = sum(h.*signal,1) + 10^(-Eb_N0_dB(ii)/20)*n; %signal received without
Beamforming
y2 = sum(bf_ch.*signal,1) + 10^(-Eb_N0_dB(ii)/20)*n; %receivd with Beamforming
% Receiver side-Equalization
y1Hat = y1./sum(h,1);
y2Hat = y2./sum(bf_ch,1);
% demodulation
y_re1 = real(y1Hat); % real part
y_im1 = imag(y1Hat); % imaginary part
y_re2 = real(y2Hat); % real part
y_im2 = imag(y2Hat); % imaginary part
%keep in mind that symbols used for M-QAM are odd numbers
%...1,3,5,7,9,13.....
% rounding to the nearest symbol
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_re1 = 2*floor(y_re1/2)+1;
ipHat_re1(find(ipHat_re1>max(alphaMqam))) = max(alphaMqam); % #ok<*FNDSB>
ipHat_re1(find(ipHat_re1<min(alphaMqam))) = min(alphaMqam);
% rounding to the nearest symbol
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_re2 = 2*floor(y_re2/2)+1;
ipHat_re2(find(ipHat_re2>max(alphaMqam))) = max(alphaMqam);
ipHat_re2(find(ipHat_re2<min(alphaMqam))) = min(alphaMqam);
% rounding to the nearest alphabet
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_im1 = 2*floor(y_im1/2)+1;
ipHat_im1(find(ipHat_im1>max(alphaMqam))) = max(alphaMqam);
ipHat_im1(find(ipHat_im1<min(alphaMqam))) = min(alphaMqam);
% rounding to the nearest alphabet
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_im2 = 2*floor(y_im2/2)+1;
ipHat_im2(find(ipHat_im2>max(alphaMqam))) = max(alphaMqam);
ipHat_im2(find(ipHat_im2<min(alphaMqam))) = min(alphaMqam);
ipHat1 = ipHat_re1 + j*ipHat_im1;
nErr1(ii) = size(find([ip- ipHat1]),2); % #ok<*AGROW> % counting the number of errors
ipHat2 = ipHat_re2 + j*ipHat_im2;

```

```

    nErr2(ii) = size(find([ip- ipHat2]),2); % counting the number of errors
end
simBer1 = nErr1/N; %BER (without beam forming)
simBer2 = nErr2/N; %BER (with beam forming)
% close all
% figure
axes(handles.BEAMFORMING);% Axis BEAMFORMING donde se realizará la gráfica en
axis
semilogy(Eb_N0_dB,simBer1,'rx-','LineWidth',2);
hold on
semilogy(Eb_N0_dB,simBer2,'bx-','LineWidth',2);
axis auto
grid on
legend('(no Beamforming-sim)','(Beamforming-sim)');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BER for 64QAM modulation in Rayleigh multitap channel');
msgbox('Beamforming Technique implemented Successfully!', 'Result');

% *****
% *****
% *****
% *****

% Traza el diagrama de la constelación con ruido de canal AWGN. con fondo
% blanco y constelaciones en azul
% h = scatterplot(s);
% hold on
% scatterplot([],[],'r*',h)
% grid
% hold off

% _____
%           TASA DE ERROR DE SIMBOLO (SER)
% _____

% Demodular la señal recibida usando la genqamdemodfunción. Determine el número de
% errores de símbolo
% y la tasa de error de símbolo.

% X=DATA
% moddata txsig
y = qammod(data,M,'bin','InputType','bit','OutputDataType', ...
    numerictype(1,M,10));

%   sin ruido
demodData = qamdemod(y,M,'bin','OutputType','bit');
[numErrors1,ser1] = symerr(data,demodData) %#ok< *ASGLU,*NOPTS>

```

```

% con ruido de canal AWGN
%rxSig = awgn(txSig,25,'measured');
demodData = qamdemod(txSig,M,'bin','OutputType','bit');
% s = isequal(data,double(demodData)) %#ok<*NOPTS>
[numErrors,ser] = symerr(data,demodData) %#ok<*NOPTS>

elseif (M_QAM==128) && (banda==5) &&
(datar==13)% _____
_____128
%-----
%-----128-QAM-----
%
%-----
%-----
% hold off;
cla(handles.constelacion,'reset') % Limpiar grafica
cla(handles.AWGN,'reset') % Limpiar grafica
cla(handles.BEAMFORMING,'reset') % Limpiar grafica
% *****
% *****
% Modulación de amplitud en cuadratura M-QAM
% *****
% *****
% Modular una secuencia de bits utilizando 64-QAM. Pase la señal a través de un canal
ruidoso.
% Muestre el diagrama de constelación resultante.

% Establezca el orden de modulación y determine el número de bits por símbolo.
M = 128;
k = log2 (M);
% numSymPerFrame = 1555; % Número de símbolos QAM por trama
% Cree una secuencia de datos binarios aleatorios. Cuando se utilizan entradas binarias, el
número de filas
% en la entrada debe ser un múltiplo entero del número de bits por símbolo.
data = randi([0 1],numSymPerFrame*k,1);% 1000

% Modular la señal utilizando entradas de bits y configurarla para que tenga una potencia
media unitaria.
txSig = qammod(data,M,'InputType','bit','UnitAveragePower',true);

% Pase la señal a través de un canal ruidoso AWGN.
% Pase la señal a través de un canal AWGN que tenga una relación señal-ruido (SNR) de 25
dB.
snrdB = 30;
rxSig = awgn (txSig, snrdB,'measured');

% _____
% TRAZO DE DIAGRAMA DE CONSTELACION

```

```

%
% Traza el diagrama de la constelación de señales moduladas usando la función scatterplot.
% axes(handles.constelacion)
hold on;
scatterplot(txSig);
saveas(gcf,'Constelacion.png')% Guardar figura como imagen
hold off;
title('128-QAM ')% Título para el diagrama de constelacion

axes(handles.constelacion);%Mostrar grafica diagrama de dispersion guardado
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;

% Traza el diagrama de la constelación con ruido de canal AWGN.
% cd = comm.ConstellationDiagram('ShowReferenceConstellation',false);
% step(cd,rxSig)%Ejecutar algoritmo de objetos del sistema
% Traza el diagrama de la constelación con ruido de canal AWGN. con fondo
% blanco y constelaciones en azul
scatterplot(rxSig); %#ok<*NASGU>
saveas(gcf,'AWGN.png')% Guardar figura como imagen
hold on
%scatterplot([],[],'r*',h)
grid
hold off
axes(handles.AWGN);%Mostrar grafica diagrama de dispersion guardado
path = 'AWGN.png';
img = imread(path);
imshow(img);
axis off;

% *****
% *****
%
%          BER Beamforming usando modulación M-QAM
% *****
% *****
% Beamforming usando modulación M-QAM en el canal de desvanecimiento Rayleigh
N = 9^6; % number of bits
%M=64; %M-QAM [M=4,16,64,128...]
k = sqrt(1/((2/3)*(M-1))); % normalizing factor
m = [1:sqrt(M)/2]; %#ok<*NBRAK>
alphaMqam = [-(2*m-1) 2*m-1]; % symbols for M-QAM modulation
Eb_N0_dB = [0:25]; % multiple Eb/N0 values
ipHat = zeros(1,N); % initialzng
% Th=3%#ok<*NOPRT> %input('Enter the number of Channel Taps :');%take input from
user(enter 5 )
for ii = 1:length(Eb_N0_dB)

```

```

ip = randsrc(1,N,alphaMqam) + j*randsrc(1,N,alphaMqam);%generate input symbols
s = k*ip; % normalization of energy to 1
n = 1/sqrt(2)*[randn(1,N) + j*randn(1,N)]; %#ok<*IJCL> % white gaussian noise, 0dB
variance
h = 1/sqrt(2)*[randn(Th,N) + j*randn(Th,N)]; % Rayleigh fading channel
signal = (1/sqrt(Th))*kron(ones(Th,1),s); %signal
% Channel and noise addition
phase= exp(j*angle(h));%calculate the phase of channel
phase_conjugate=conj(phase);%conjugate of phase
bf_ch= h.*phase_conjugate;%beamforming channel
y1 = sum(h.*signal,1) + 10^(-Eb_N0_dB(ii)/20)*n; %signal received without
Beamforming
y2 = sum(bf_ch.*signal,1) + 10^(-Eb_N0_dB(ii)/20)*n; %receivd with Beamforming
% Receiver side-Equalization
y1Hat = y1./sum(h,1);
y2Hat = y2./sum(bf_ch,1);
% demodulation
y_re1 = real(y1Hat); % real part
y_im1 = imag(y1Hat); % imaginary part
y_re2 = real(y2Hat); % real part
y_im2 = imag(y2Hat); % imaginary part
%keep in mind that symbols used for M-QAM are odd numbers
%...1,3,5,7,9,13.....
% rounding to the nearest symbol
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_re1 = 2*floor(y_re1/2)+1;
ipHat_re1(find(ipHat_re1>max(alphaMqam))) = max(alphaMqam); %#ok<*FNDSB>
ipHat_re1(find(ipHat_re1<min(alphaMqam))) = min(alphaMqam);
% rounding to the nearest symbol
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_re2 = 2*floor(y_re2/2)+1;
ipHat_re2(find(ipHat_re2>max(alphaMqam))) = max(alphaMqam);
ipHat_re2(find(ipHat_re2<min(alphaMqam))) = min(alphaMqam);
% rounding to the nearest alphabet
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_im1 = 2*floor(y_im1/2)+1;
ipHat_im1(find(ipHat_im1>max(alphaMqam))) = max(alphaMqam);
ipHat_im1(find(ipHat_im1<min(alphaMqam))) = min(alphaMqam);
% rounding to the nearest alphabet
% 0 to 2 --> 1
% 2 to 4 --> 3
% 4 to 6 --> 5 etc
ipHat_im2 = 2*floor(y_im2/2)+1;
ipHat_im2(find(ipHat_im2>max(alphaMqam))) = max(alphaMqam);

```

```

ipHat_im2(find(ipHat_im2<min(alphaMqam))) = min(alphaMqam);
ipHat1 = ipHat_re1 + j*ipHat_im1;
nErr1(ii) = size(find([ip- ipHat1]),2); %#ok<*AGROW> % counting the number of errors
ipHat2 = ipHat_re2+ j*ipHat_im2;
nErr2(ii) = size(find([ip- ipHat2]),2); % counting the number of errors
end
simBer1 = nErr1/N; %BER (without beam forming)
simBer2 = nErr2/N; %BER (with beam forming)
% close all
% figure
axes(handles.BEAMFORMING);%Axis BEAMFORMING donde se realizará la gráfica en
axis
semilogy(Eb_N0_dB,simBer1,'rx-','LineWidth',2);
hold on
semilogy(Eb_N0_dB,simBer2,'bx-','LineWidth',2);
axis auto
grid on
legend('no Beamforming-sim'),('Beamforming-sim');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BER for 128QAM modulation in Rayleigh multitap channel');
msgbox("'Beamforming Technique implemented Successfully!'",'Result');

% *****
% *****
% *****
% *****

% Traza el diagrama de la constelación con ruido de canal AWGN. con fondo
% blanco y constelaciones en azul
% h = scatterplot(s);
% hold on
% scatterplot([],[],'r*',h)
% grid
% hold off

% _____
%
% TASA DE ERROR DE SIMBOLO (SER)
% _____

% Demodular la señal recibida usando la genqamdemodfunción. Determine el número de
% errores de símbolo
% y la tasa de error de símbolo.

% X=DATA
% moddata txsig
y = qammod(data,M,'bin','InputType','bit','OutputDataType', ...
    numerictype(1,M,10));

```

```

% sin ruido
demodData = qamdemod(y,M,'bin','OutputType','bit');
[numErrors1,ser1] = symerr(data,demodData) %#ok<*ASGLU,*NOPTS>

% con ruido de canal AWGN
% rxSig = awgn(txSig,25,'measured');
demodData = qamdemod(txSig,M,'bin','OutputType','bit');
% s = isequal(data,double(demodData)) %#ok<*NOPTS>
[numErrors,ser] = symerr(data,demodData) %#ok<*NOPTS>
elseif (M_QAM==256) && (banda==5) && (datar==13 || datar==234 || datar==347 ||
datar==69)%
_____
_____256
%-----
%-----256-QAM-----
%
%-----
%-----
% hold off;
cla(handles.constelacion,'reset') % Limpiar grafica
cla(handles.AWGN,'reset') % Limpiar grafica
cla(handles.BEAMFORMING,'reset') % Limpiar grafica
% *****
% *****
% Modulación de amplitud en cuadratura M-QAM
% *****
% *****
% Modular una secuencia de bits utilizando 64-QAM. Pase la señal a través de un canal
ruidoso.
% Muestre el diagrama de constelación resultante.

% Establezca el orden de modulación y determine el número de bits por símbolo.
M = 256;
k = log2 (M);
% numSymPerFrame = 10000; % Número de símbolos QAM por trama
% Cree una secuencia de datos binarios aleatorios. Cuando se utilizan entradas binarias, el
número de filas
% en la entrada debe ser un múltiplo entero del número de bits por símbolo.
data = randi([0 1],numSymPerFrame*k,1);% 1000

% Modular la señal utilizando entradas de bits y configurarla para que tenga una potencia
media unitaria.
txSig = qammod(data,M,'InputType','bit','UnitAveragePower',true);

% Pase la señal a través de un canal ruidoso AWGN.
% Pase la señal a través de un canal AWGN que tenga una relación señal-ruido (SNR) de 25
dB.
snrdB = 30;
rxSig = awgn (txSig, snrdB,'measured');

```

```

%
%
% TRAZO DE DIAGRAMA DE CONSTELACION
%
% Traza el diagrama de la constelación de señales moduladas usando la función scatterplot.
% axes(handles.constelacion)
hold on;
scatterplot(txSig);
saveas(gcf,'Constelacion.png')% Guardar figura como imagen
hold off;
title('256-QAM ' )% Título para el diagrama de constelacion

axes(handles.constelacion);%Mostrar grafica diagrama de dispersion guardado
path = 'Constelacion.png';
img = imread(path);
imshow(img);
axis off;

% Traza el diagrama de la constelación con ruido de canal AWGN.
% cd = comm.ConstellationDiagram('ShowReferenceConstellation',false);
% step(cd,rxSig)%Ejecutar algoritmo de objetos del sistema
% Traza el diagrama de la constelación con ruido de canal AWGN. con fondo
% blanco y constelaciones en azul
scatterplot(rxSig); %#ok<*NASGU>
saveas(gcf,'AWGN.png')% Guardar figura como imagen
hold on
%scatterplot([],[],'r*',h)
grid
hold off
axes(handles.AWGN);%Mostrar grafica diagrama de dispersion guardado
path = 'AWGN.png';
img = imread(path);
imshow(img);
axis off;

% *****
% *****
% BER Beamforming usando modulación M-QAM
% *****
% *****
% Beamforming usando modulación M-QAM en el canal de desvanecimiento Rayleigh
N = 9^6; % number of bits
M=64; %M-QAM [M=4,16,63..]
k = sqrt(1/((2/3)*(M-1))); % normalizing factor
m = [1:sqrt(M)/2]; %#ok<*NBRAK>
alphaMqam = [-(2*m-1) 2*m-1]; % symbols for M-QAM modulation
Eb_N0_dB = [0:25]; % multiple Eb/N0 values
ipHat = zeros(1,N); % inicialzng

```

```

% Th=3%#ok<*NOPRT> %input('Enter the number of Channel Taps :');%take input from
user(enter 5 )
for ii = 1:length(Eb_N0_dB)
    ip = randsrc(1,N,alphaMqam) + j*randsrc(1,N,alphaMqam);%generate input symbols
    s = k*ip; % normalization of energy to 1
    n = 1/sqrt(2)*[randn(1,N) + j*randn(1,N)]; %#ok<*IJCL> % white gaussian noise, 0dB
variance
    h = 1/sqrt(2)*[randn(Th,N) + j*randn(Th,N)]; % Rayleigh fading channel
    signal = (1/sqrt(Th))*kron(ones(Th,1),s); %signal
    % Channel and noise addition
    phase = exp(j*angle(h));%calculate the phase of channel
    phase_conjugate = conj(phase);%conjugate of phase
    bf_ch = h.*phase_conjugate;%beamforming channel
    y1 = sum(h.*signal,1) + 10^(-Eb_N0_dB(ii)/20)*n; %signal received without
Beamforming
    y2 = sum(bf_ch.*signal,1) + 10^(-Eb_N0_dB(ii)/20)*n; %receivd with Beamforming
    % Receiver side-Equalization
    y1Hat = y1./sum(h,1);
    y2Hat = y2./sum(bf_ch,1);
    % demodulation
    y_re1 = real(y1Hat); % real part
    y_im1 = imag(y1Hat); % imaginary part
    y_re2 = real(y2Hat); % real part
    y_im2 = imag(y2Hat); % imaginary part
    %keep in mind that symbols used for M-QAM are odd numbers
    %...1,3,5,7,9,13.....
    % rounding to the nearest symbol
    % 0 to 2 --> 1
    % 2 to 4 --> 3
    % 4 to 6 --> 5 etc
    ipHat_re1 = 2*floor(y_re1/2)+1;
    ipHat_re1(find(ipHat_re1>max(alphaMqam))) = max(alphaMqam); %#ok<*FNDSB>
    ipHat_re1(find(ipHat_re1<min(alphaMqam))) = min(alphaMqam);
    % rounding to the nearest symbol
    % 0 to 2 --> 1
    % 2 to 4 --> 3
    % 4 to 6 --> 5 etc
    ipHat_re2 = 2*floor(y_re2/2)+1;
    ipHat_re2(find(ipHat_re2>max(alphaMqam))) = max(alphaMqam);
    ipHat_re2(find(ipHat_re2<min(alphaMqam))) = min(alphaMqam);
    % rounding to the nearest alphabet
    % 0 to 2 --> 1
    % 2 to 4 --> 3
    % 4 to 6 --> 5 etc
    ipHat_im1 = 2*floor(y_im1/2)+1;
    ipHat_im1(find(ipHat_im1>max(alphaMqam))) = max(alphaMqam);
    ipHat_im1(find(ipHat_im1<min(alphaMqam))) = min(alphaMqam);
    % rounding to the nearest alphabet
    % 0 to 2 --> 1
    % 2 to 4 --> 3

```

```

% 4 to 6 --> 5 etc
ipHat_im2 = 2*floor(y_im2/2)+1;
ipHat_im2(find(ipHat_im2>max(alphaMqam))) = max(alphaMqam);
ipHat_im2(find(ipHat_im2<min(alphaMqam))) = min(alphaMqam);
ipHat1 = ipHat_re1 + j*ipHat_im1;
nErr1(ii) = size(find([ip- ipHat1]),2); %#ok<*AGROW> % counting the number of errors
ipHat2 = ipHat_re2+ j*ipHat_im2;
nErr2(ii) = size(find([ip- ipHat2]),2); % counting the number of errors
end
simBer1 = nErr1/N; %BER (without beam forming)
simBer2 = nErr2/N; %BER (with beam forming)
% close all
% figure
axes(handles.BEAMFORMING);%Axis BEAMFORMING donde se realizará la gráfica en
axis
semilogy(Eb_N0_dB,simBer1,'rx-','LineWidth',2);
hold on
semilogy(Eb_N0_dB,simBer2,'bx-','LineWidth',2);
axis auto
grid on
legend('(no Beamforming-sim)','(Beamforming-sim)');
xlabel('Eb/No, dB');
ylabel('Bit Error Rate');
title('BER for 256QAM modulation in Rayleigh multitap channel');
msgbox('"Beamforming Technique implemented Successfully!"','Result');

% *****
% *****
% *****
% *****

% Traza el diagrama de la constelación con ruido de canal AWGN. con fondo
% blanco y constelaciones en azul
% h = scatterplot(s);
% hold on
% scatterplot([],[],'r*',h)
% grid
% hold off

% _____
% TASA DE ERROR DE SIMBOLO (SER)
% _____

% Demodular la señal recibida usando la genqamdemodfunción. Determine el número de
% errores de símbolo
% y la tasa de error de símbolo.

% X=DATA

```

```

%moddata txsig
y = qammod(data,M,'bin','InputType','bit','OutputDataType', ...
    numerictype(1,M,10));

%   sin ruido
demodData = qamdemod(y,M,'bin','OutputType','bit');
[numErrors1,ser1] = symerr(data,demodData) %#ok<*ASGLU,*NOPTS>

%   con ruido de canal AWGN
%rxSig = awgn(txSig,25,'measured');
demodData = qamdemod(txSig,M,'bin','OutputType','bit');
%s = isequal(data,double(demodData)) %#ok<*NOPTS>
[numErrors,ser] = symerr(data,demodData) %#ok<*NOPTS>

else
    %msgbox('Error !','ERROR','error');
    errordlg('Error Por favor revise la tabla e intente nuevamente!','ERROR DE SELECCIÓN ')
end
end
else
    errordlg('Error El número de canales y simbolos debe ser mayor a cero!','ERROR
NÚMERO DE CH-SYM ')
end

% --- Executes on button press in Salir.
function Salir_Callback(hObject, eventdata, handles)
% hObject   handle to Salir (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)
a = questdlg('¿Desea salir del programa?','Salir','Si','No','No');
if strcmp(a,'No')
    return;
end
clc;
clear all; %#ok<*CLALL>
close all

% --- Executes during object creation, after setting all properties.
function trama_CreateFcn(hObject, eventdata, handles)
% hObject   handle to trama (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%   See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

end

```
% --- Executes during object creation, after setting all properties.
function canales_CreateFcn(hObject, eventdata, handles)
% hObject    handle to canales (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: edit controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on selection change in datarate.
function datarate_Callback(hObject, eventdata, handles)
% hObject    handle to datarate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Hints: contents = cellstr(get(hObject,'String')) returns datarate contents as cell array
%     contents{get(hObject,'Value')} returns selected item from datarate
```

```
global datar %#ok<*NUSED>
handles.cod=get(hObject,'Value');
```

```
switch handles.cod
    case 1
        datar=450; %450Mps
    case 2
        datar=600;%600Mbps
    case 3
        datar=13;%1.3Gps
    case 4
        datar=234;%2.34Gps
    case 5
        datar=347;%3.47Gps
    case 6
        datar=69;%6.9Gps
end
```

```
% --- Executes during object creation, after setting all properties.
```

```
function datarate_CreateFcn(hObject, eventdata, handles)
% hObject    handle to datarate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% Hint: popupmenu controls usually have a white background on Windows.
%     See ISPC and COMPUTER.
```

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

% --- Executes on selection change in banda.

```

function banda_Callback(hObject, eventdata, handles)
% hObject    handle to banda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

% Hints: contents = cellstr(get(hObject,'String')) returns banda contents as cell array
% contents{get(hObject,'Value')} returns selected item from banda

```

global banda %#ok<*NUSED>
handles.cod=get(hObject,'Value');
switch handles.cod
    case 1
        banda=24;% 2.4GHz
    case 2
        banda=5; % 5GHz
end

```

% --- Executes during object creation, after setting all properties.

```

function banda_CreateFcn(hObject, eventdata, handles)
% hObject    handle to banda (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

% Hint: popupmenu controls usually have a white background on Windows.
% See ISPC and COMPUTER.

```

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

% --- Executes on selection change in select_code.

```

function select_code_Callback(hObject, eventdata, handles) %#ok<*INUSL,*DEFNU>
% hObject    handle to select_code (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global M_QAM %#ok<*NUSED>
handles.cod=get(hObject,'Value');
switch handles.cod
    case 1
        M_QAM=8;
        set(handles.banda,'visible','off');
        set(handles.canales,'visible','off');
        set(handles.datarate,'visible','off');

```

```

set(handles.uibuttongroup4,'visible','off');
set(handles.uibuttongroup5,'visible','off');
set(handles.uibuttongroup6,'visible','off');
set(handles.uibuttongroup7,'visible','off');
set(handles.uipanel1,'visible','on');
set(handles.uipanel2,'visible','on');
set(handles.uipanel3,'visible','on');

```

case 2

```

M_QAM=16;
set(handles.uipanel1,'visible','on');
set(handles.uipanel2,'visible','on');
set(handles.uipanel3,'visible','on');
set(handles.banda,'visible','off');
set(handles.canales,'visible','off');
set(handles.datarate,'visible','off');
set(handles.uibuttongroup4,'visible','off');
set(handles.uibuttongroup5,'visible','off');
set(handles.uibuttongroup6,'visible','off');
set(handles.uibuttongroup7,'visible','off');

```

case 3

```

M_QAM=32;
set(handles.uipanel2,'visible','on');
set(handles.uipanel3,'visible','on');
set(handles.uipanel1,'visible','on');
set(handles.banda,'visible','off');
set(handles.canales,'visible','off');
set(handles.datarate,'visible','off');
set(handles.uibuttongroup4,'visible','off');
set(handles.uibuttongroup5,'visible','off');
set(handles.uibuttongroup6,'visible','off');
set(handles.uibuttongroup7,'visible','off');

```

case 4

```

M_QAM=64;
set(handles.uipanel3,'visible','off');
set(handles.uipanel2,'visible','off');
set(handles.uipanel1,'visible','off');
set(handles.banda,'visible','on');
set(handles.canales,'visible','on');
set(handles.trama,'visible','on');
set(handles.datarate,'visible','on');
set(handles.uibuttongroup4,'visible','on');
set(handles.uibuttongroup5,'visible','on');
set(handles.uibuttongroup6,'visible','on');
set(handles.uibuttongroup7,'visible','on');

```

end

```

% Hints: contents = cellstr(get(hObject,'String')) returns select_code contents as cell array
%     contents{get(hObject,'Value')} returns selected item from select_code

```

```

% --- Executes during object creation, after setting all properties.

```

```
function select_code_CreateFcn(hObject, eventdata, handles) %#ok<*INUSD>
% hObject   handle to select_code (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

*Anexo 6 Guía Didáctica de Comunicación digital.***PRÁCTICA #1****Transmisor de Modulación QPSK**

LABORATORIO DE	Comunicación Digital	DURACIÓN
INTEGRANTES:	• _____	2 horas

1	OBJETIVOS
	<ul style="list-style-type: none"> • Adquirir conocimientos sobre los conceptos básicos de la modulación PSK. • Configurar el software Gnu Radio para la transmisión con el SDR. • Interpretar y analizar los resultados obtenidos.

2	EQUIPOS NECESARIOS
	<p>Para el desarrollo de esta práctica es necesario la utilización de los siguientes equipos:</p> <ul style="list-style-type: none"> ✓ Computador ✓ SDR HackRF One ✓ Software Gnu Radio

3	FUNDAMENTO TEORICO
	<p style="text-align: center;">Modulación Digital</p> <p>Las modulaciones digitales implican el procesamiento de algunos parámetros en una onda portadora en función de una señal conocida como onda moduladora, que es una señal</p>

digital binaria de "1" y "0" que transporta información, para que la señal pueda ser transmitida por un canal [1].

Modulación Por Desplazamiento De Fase En Cuadratura QPSK (4-PSK)

En cada bit está encriptado dividido en parejas dándoles el valor nominal de la decoración como cambio de etapa en comparación con la señal anterior. La fase de la señal se modula para codificar bits de información digital en cada cambio de fase [2].

$$N = \text{Log}_2(M)$$

Donde:

Ecuación 1

N= Número de bits codificados

M= Cantidad de condiciones

La fase de la señal portadora "N" puede tener un valor en serie distanciados en un ángulo $\theta = \frac{2\pi}{N}$. En el caso N = 4, tiene 4-PSK o QPSK.

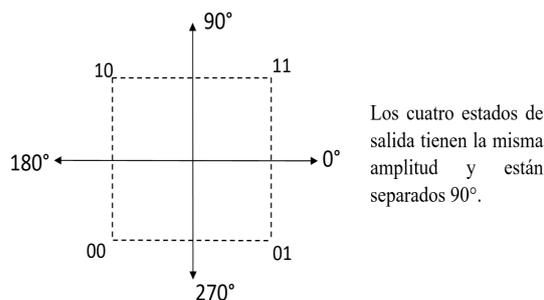
Codificación Por Cambio De Fase En Cuadratura

Codificación de compensación de fase cuadrática. En la **Figura. 6** representa las cuatro fases igualmente espaciadas: 00, 01, 11, 10, transmisión QPSK es de dos bits por fase el doble de eficiencia de ancho de banda Rango de frecuencia de BPSK. [3]

Figura. 1 Entrada y salida QPSK

Entrada binaria		Fase de salida de QPSK
Q	I	
0	0	-135°
0	1	-45°

Figura. 2 Constelación QPSK



1	0	+135°
1	1	+45°

Nota. Estado de salida de la modulación QPSK. [3]

Hardware Para SDR

HackRF One es un dispositivo de radio definido por software capaz de transmitir o recibir Señales de radio de 1MHz a 6GHz fabricadas por Great Scott Gadgets [4].

Figura. 3 HackRF One.



Nota. Elaborador por el autor.

Característica	Valor
Rango de frecuencias	1MHz-6GHz
Ancho de banda RF	20MHz
Profundidad de muestreo	8 bits
Velocidad de muestreo	20MSPS
Canales del transmisor	1
Receptores	1
Dúplex	Semidúplex
Interfaz	USB 2.0
Puertas lógicas programables	USB 2.0
Chipset	MAX5864, MAX2837, RFFC5072
Código abierto	Código abierto completo
Potencia de transmisión	-10dBm+ (15dBm @ 2.4GHz)
Compatible	Windows, Linux, SDR y GNU Radio.

Tabla 1 Características del HackRF One [5]

Gnu Radio

Gnu Radio es un programa de código abierto diseñado para crear bloques a fin de implementar sistemas SDR, y cuenta con un entorno de simulación totalmente especializado para recrear ambientes óptimos en futuras comunicaciones inalámbricas. [6]

4

DESARROLLO

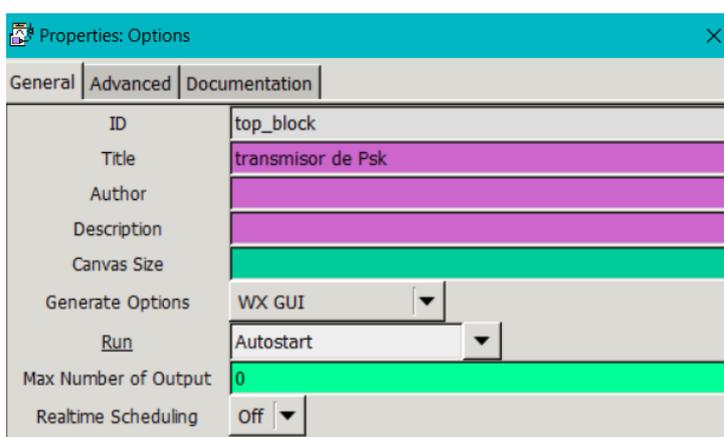
TRANSMISOR De Modulación QPSK

Paso 1: Se ingresa al programa **GNU Radio Companion** (GRC).

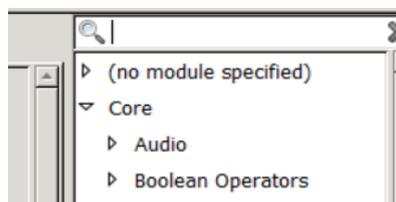


Paso 2: En la ventana principal se configura el bloque de **Options en WX GUI**.

Options
ID: top_block
Title: transmisor PSK
Generate Options: WX GUI

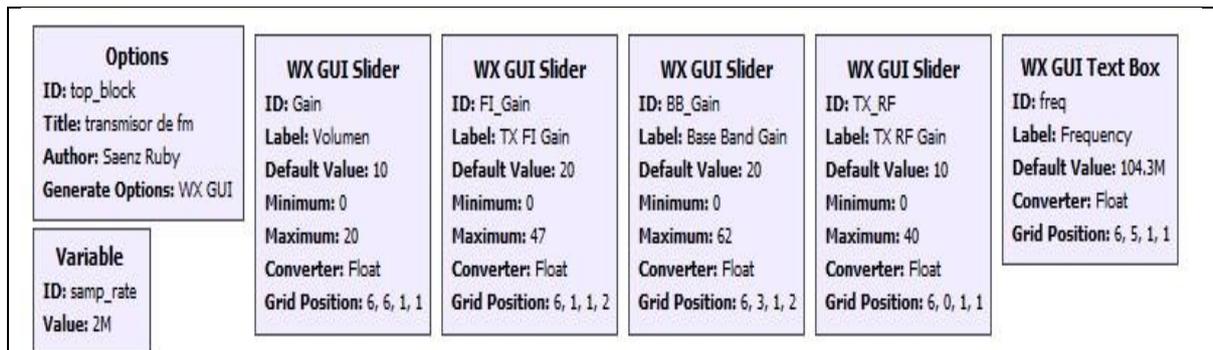


Paso 3: Para buscar los bloques que se mencionan a continuación, se dirige a la lupa que esta la barra de herramientas.



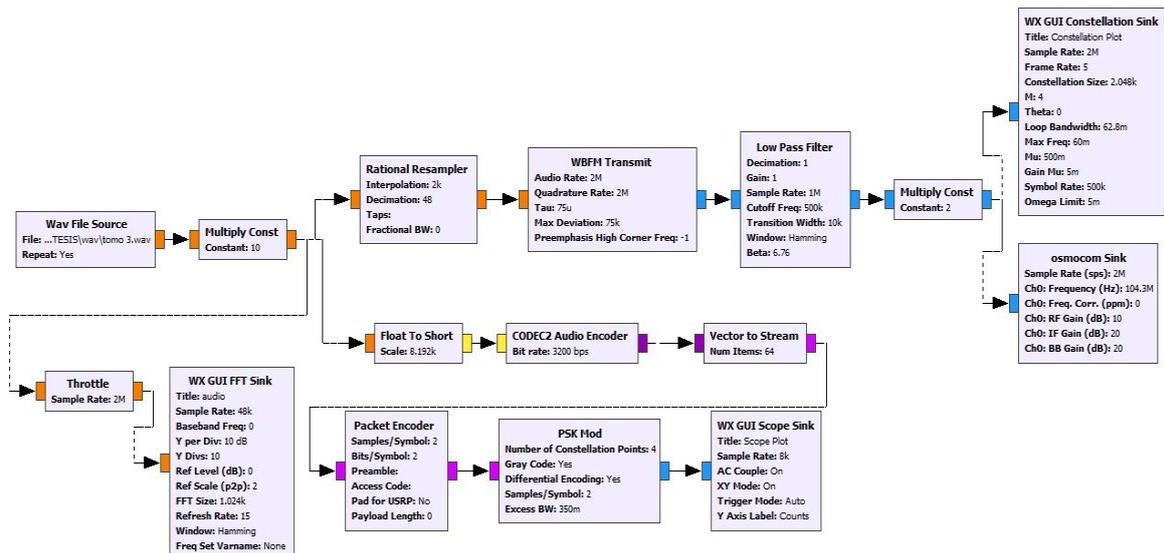
Paso 4: Configuración las variables y bloques:

VARIABLES



- **Gain:** Es el volumen de la señal moduladora con un valor min de 0, valor máx. de 20 y un valor determinado de 10.
- **TX FI Gain:** Es la ganancia de frecuencia intermedia de transmisión con un valor min de 0, valor máx. de 47 y un valor determinado de 20.
- **BB_Gain: Basa Banda Gain:** es la ganancia de banda base con un valor min de 0, valor máx de 62 y un valor determinado de 20.
- **TX RF GAIN:** ganancia de transmisión de radio frecuencia con un valor min de 0, valor máx de 40 y un valor determinado de 10.
- **Freq:** permite escribir el valor de la frecuencia en la cual se quiere transmitir.

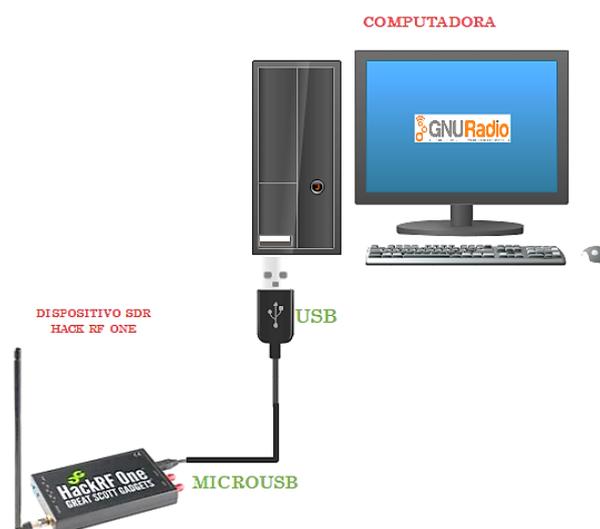
DESCRIPCIÓN DE BLOQUES



- **Wav File Source:** Lector de archivos WAV.
- **Multiply Const:** Sirve para multiplicar la señal de entrada por un valor constante, amplificando la señal sin afectarla.
- **Throttle:** Regulador de flujo de datos. Este bloque mantiene una tasa constante de datos en su salida.
- **Float To Char:** Este conversor de tipo de datos funciona como cuantizador de 8.192k niveles.

- **CODEC2 Audio Encoder:** diseñado para ofrecer compresión de voz de baja tasa de bits con buena calidad de audio.
- **Vector to Stream:** Se utiliza para convertir un vector de datos en una secuencia de datos de flujo continuo.
- **Packet Encoder:** Codificador de paquetes. Se encarga de fragmentar en cada dato tipo char a la salida un número de bits indicado por la variable bits/sym de cada byte que ingresa. La variable samp/sym se encarga de entregar ese número de muestras por símbolo de entrada, en este caso se están duplicando los símbolos que ingresan.
- **PSK Mod:** Permite especificar el número de fases, la tasa de símbolos.
- **WX GUI Scope Sink:** es un bloque de visualización que se utiliza para mostrar gráficamente señales en tiempo real.
- **Rational Resampler:** Muestreador racional. Este bloque permite cambiar entre tasas de muestreo enteras, recibiendo por muestras (decimation) y arrojando y muestras (interpolation), con x e y números naturales.
- **WBFM Transmit:** La transmisión de FM de banda ancha (WBFM - Wideband Frequency Modulation) Toma un solo flujo de entrada flotante de muestras de audio en el rango $[-1,+1]$ y produce una sola salida de banda base compleja modulada FM.
- **Low Pass Filter:** Filtro de paso bajo. Las muestras son filtradas antes de ser ingresadas al osocom, con el fin de garantizar que ocuparán el ancho de banda que se desea, que en este caso es la mitad de la tasa de símbolos del osocom.

Paso 5: Se debe conectar el HackRF One al USB del CPU de la computadora.



Paso 6: Se cargan los bloques de código al SDR.



Paso 7: Por último, se ejecuta.

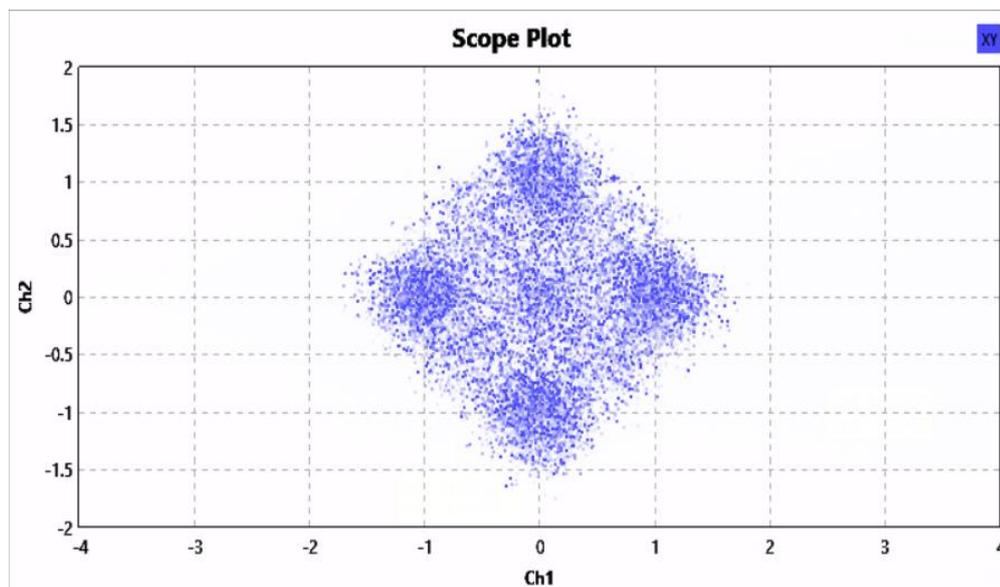


6

RESULTADOS**Simulación**

Como se puede ver en la figura, las características del gráfico de constelación están conformado con los siguientes valores 10 dBm de TX RF Gain, 20 dB de TX FI Gain, y 20 dB de Base Band Gain están proporcionando una señal de audio amplificada en la banda de frecuencia de 104.3 MHz.

Se puede observar que en la modulación QPSK, las constelaciones se encuentran difusas o borrosas esto se debe al ruido presentado en la transmisión debido a diversos factores tales como: interferencia de electromagnética, ruido en el canal de transmisión proveniente de dispositivos electrónicos cercanos, atenuación de la señal o problemas en la modulación de la señal.

**Cálculos teóricos**

$$SNR_{QPSK}(dB) = 10 \log_{10} \left(\frac{P_{Tx}}{P_{Ruido}} \right)$$

Donde:

P_{Tx} = Es la potencia de la señal transmitida (sin ruido).

P_{Rx} = Potencia de la señal recibida (con ruido).

Datos:

$$P_{Tx} = 792u$$

$$P_{Rx} = 1100u$$

$$P_{Ruido} = P_{Rx} - P_{Tx}$$

$$P_{Ruido} = 1100u - 792u = 308u$$

$$SNR_{4PSK}(dB) = 10 \log_{10} \left(\frac{792u}{308u} \right) = 4.10 \text{ dB.}$$

Como se puede observar en la figura que las constelaciones no se encuentran muy bien definidas debido a que esta transmisión ha sido afectada por un ruido de 308u, que se ha calculado a través de la diferencia de la potencia recibida y la potencia transmitida. Se visualiza que la potencia de este ruido con relación a la potencia del audio transmitido es menor, por lo que permitirá tener un SNR aceptable en la transmisión del audio esto se puede respaldar visualmente en el diagrama de constelaciones al observar los puntos un poco difusos.

7	CONCLUSIONES.
<ul style="list-style-type: none"> • Experimente con diferentes parámetros de señal PSK (como amplitud, fase o frecuencia) para ver cómo estos cambios afectan la calidad y la detección de la señal. Los resultados obtenidos se analizan en detalle para comprender mejor el comportamiento de la modulación PSK. • Registrar cuidadosamente las actividades realizadas durante el laboratorio y los resultados obtenidos. Los resultados se comparan con predicciones teóricas para validar la comprensión de los conceptos de modulación PSK. También ayudará a identificar 	

posibles inconsistencias y errores durante el experimento.

8	BIBLIOGRAFIA.
	<p>[1] J. E. S. Rodriguez, «Telecomundo,» Modulaciones Digitales, 16 Octubre 2016. [En línea]. Available: https://telecomundo.wordpress.com/modulaciones-digitales/. [Último acceso: 10 Noviembre 2023].</p> <p>[2] B. A. Forouzan, TRANSMISIÓN DE DATOS Y REDES DE COMUNICACIONES 4ED, Madrid: McGraw-Hill, 2011.</p> <p>[3] «Modulación Digital,» Sistemas de Telecomunicación, [En línea]. Available: https://www.uv.es/~hertz/hertz/Docencia/teoria/Trasmdigital.pdf. [Último acceso: 23 Agosto 2022].</p> <p>[4] Greatscottgadgets, «Greatscottgadgets.com,» 2009 - 2021. [En línea]. Available: https://greatscottgadgets.com/hackrf/one/. [Último acceso: 27 junio 2022].</p> <p>[5] Lab401, «Lab401,» 1, 2023. [En línea]. Available: https://lab401.com/es/products/hackrf-one. [Último acceso: 21 Noviembre 2023].</p> <p>[6] «Gnu Radio,» 2022. [En línea]. Available: https://www.gnuradio.org/. [Último acceso: 27 Agosto 2022].</p>

*Anexo 7 Guía Didáctica de Procesamiento de señales digitales.***PRÁCTICA #2****Receptor**

LABORATORIO	Procesamiento De Señales Digitales	DURACIÓN
INTEGRANTES:	• _____	2 horas

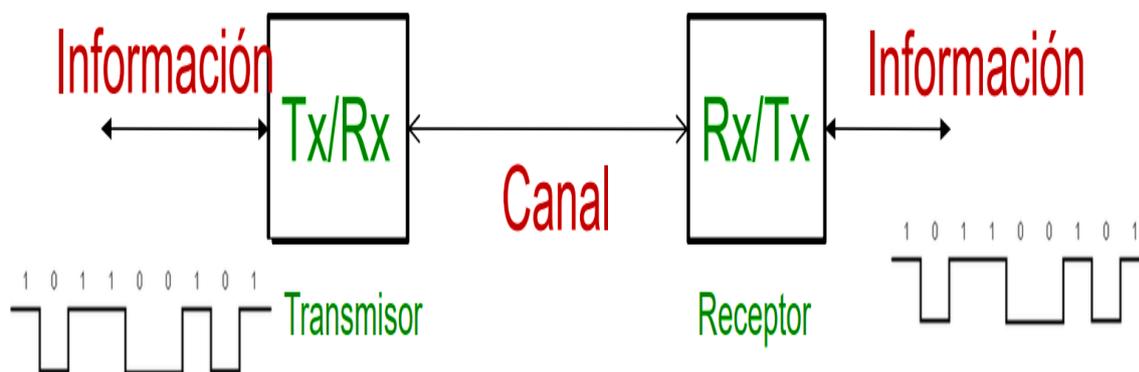
1	OBJETIVOS
	<ul style="list-style-type: none"> • Adquirir conocimientos sobre el sistema de comunicación digital. • Configurar el software Gnu Radio para la recepción con el SDR. • Interpretar y analizar los resultados obtenidos.

2	EQUIPOS NECESARIOS
	<p>Para el desarrollo de esta práctica es necesario la utilización de los siguientes equipos:</p> <ul style="list-style-type: none"> ✓ Computador ✓ SDR HackRF One ✓ Software Gnu Radio

3	FUNDAMENTO TEORICO
	Sistema de comunicación digital

El concepto básico de la comunicación digital es el traslado de información punto a punto sobre canales digitales o analógicos y puede ser representado como: El transmisor (Tx) es el que modifica la información original haciéndola adecuada para su transmisión[1], en la cual el receptor (Rx) finalmente debe convertir la información recibida a su forma original para que pueda ser enviada a su destino y a su vez procesada[2].

Figura. 101 Sistema de comunicación digital.



Nota. Se aprecia la transmisión y recepción de una señal digital que pasa por medio del canal transmitiendo información[3].

Hardware Para SDR

HackRF One es un dispositivo de radio definido por software capaz de transmitir o recibir Señales de radio de 1MHz a 6GHz fabricadas por Great Scott Gadgets [4].

Figura. 3 HackRF One.



Nota. Elaborador por el autor.

Característica	Valor
Rango de frecuencias	1MHz-6GHz
Ancho de banda RF	20MHz
Profundidad de muestreo	8 bits
Velocidad de muestreo	20MSPS
Canales del transmisor	1
Receptores	1
Dúplex	Semidúplex
Interfaz	USB 2.0
Puertas lógicas programables	USB 2.0
Chipset	MAX5864, MAX2837, RFFC5072
Código abierto	Código abierto completo
Potencia de transmisión	-10dBm+ (15dBm @ 2.4GHz)
Compatible	Windows, Linux, SDR y GNU Radio.

Tabla 1 Características del HackRF One [5]

Gnu Radio

Gnu Radio es un programa de código abierto diseñado para crear bloques a fin de implementar sistemas SDR, y cuenta con un entorno de simulación totalmente especializado para recrear ambientes óptimos en futuras comunicaciones inalámbricas. [6]

5

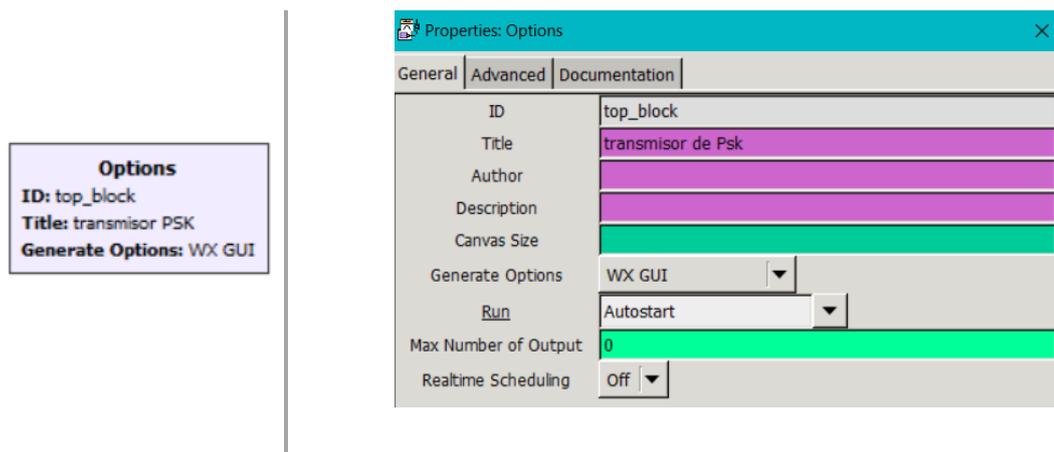
DESARROLLO

RECEPTOR

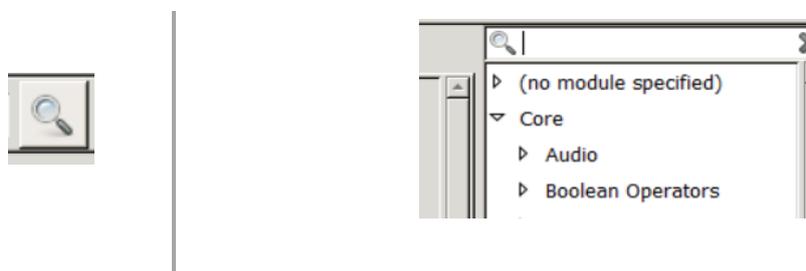
Paso 1: Se ingresa al programa **GNU Radio Companion** (GRC).



Paso 2: En las ventana principal se configura el bloque de **Options en WX GUI**.



Paso 3: Para buscar los bloques que se mencionan a continuación, se dirige a la lupa que esta la barra de herramientas.



Paso 4: Se debe configurar las variables y bloques:

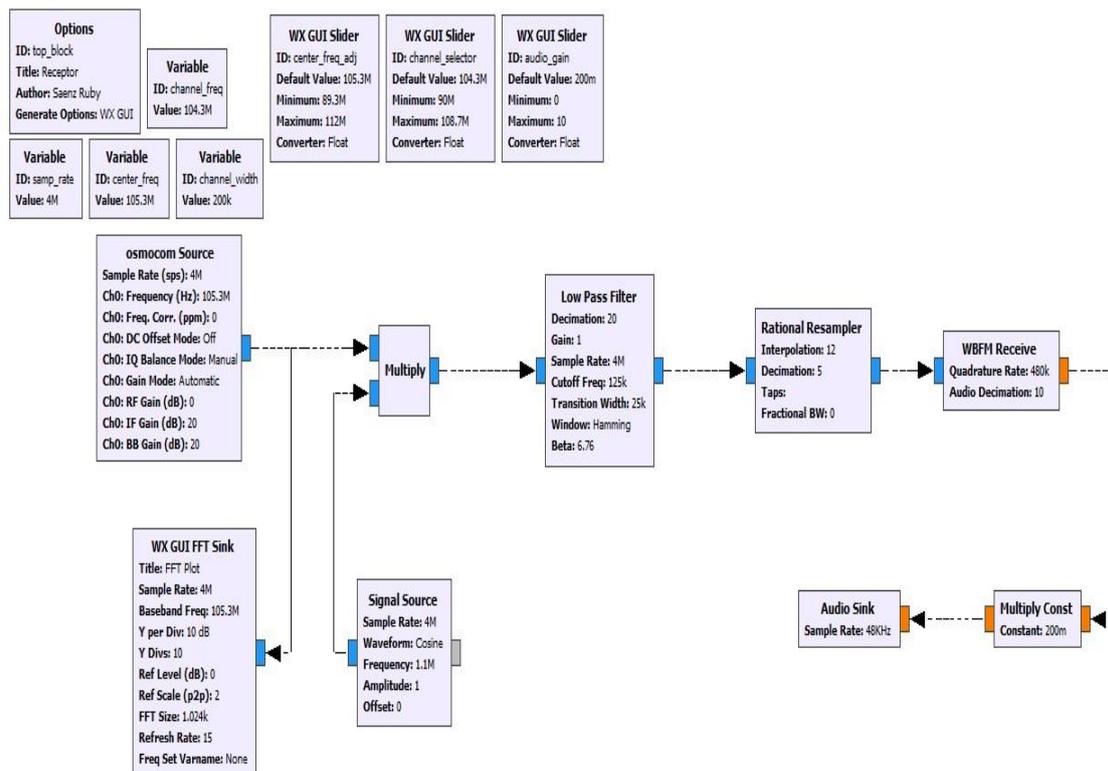
Variables

Este bloque viene por defecto el `sample_rate`, pero si se puede modificar a otro tipo de variables como se muestran a continuación:

- **Center_freq:** trabaja con una frecuencia central de 105.3 Mhz
- **Channel_freq:** la frecuencia del canal es de 104.3 Mhz
- **Channel_width:** El ancho de banda es de 200 Khz
- **WX GUI Slider:** Este bloque permite variar los valores de un mínimo a un máximo.
- **Center_freq_adj:** Ajusta de la frecuencia central de 105.3 Mhz entre 89.3 Mhz a 112 Mhz
- **Channel_selector:** Permite seleccionar las frecuencias que van desde 90 Mhz a 108.7 Mhz evaluando en 104.3 Mhz.
- **Audio_gain:** La ganancia del audio va desde 0 a 10 y se centra en 200 mhz

DESCRIPCIÓN DE BLOQUES

El diagrama del receptor de las modulaciones de PSK y QAM, sirve para verifica su correcta transmisión de las modulaciones en la frecuencia de 104.3 Mhz.



- **Osmocom Source:** Es el que permite recibir la señal mediante el SDR (HackRF One).

- **Signal Source:** El bloque permite escoger el tipo de onda con la que da una amplitud de 1 y una frecuencia de 1.1 Mhz.
- **Multiply:** Recibe el ingreso de dos señales, la señal recibida y la multiplica por una señal generada, en la cual presenta una sola señal de salida.
- **WBFM Receiver:** Este bloque demodula una señal FM con tipo de dato complex y lo convierte a float, la tasa de cuadratura con la que trabaja es de 480 Khz.
- **Audio Sink:** Este bloque permite interpretar la señal que genero el bloque signal source, cambiándolo a un lenguaje más entendible para el usuario, trabaja con una entrada float y con un frecuencia de muestreo de 48 Khz.

Paso 5: Se debe conectar el SDR al Cpu de la computadora.



Paso 6: Se cargan los bloques de código al SDR.



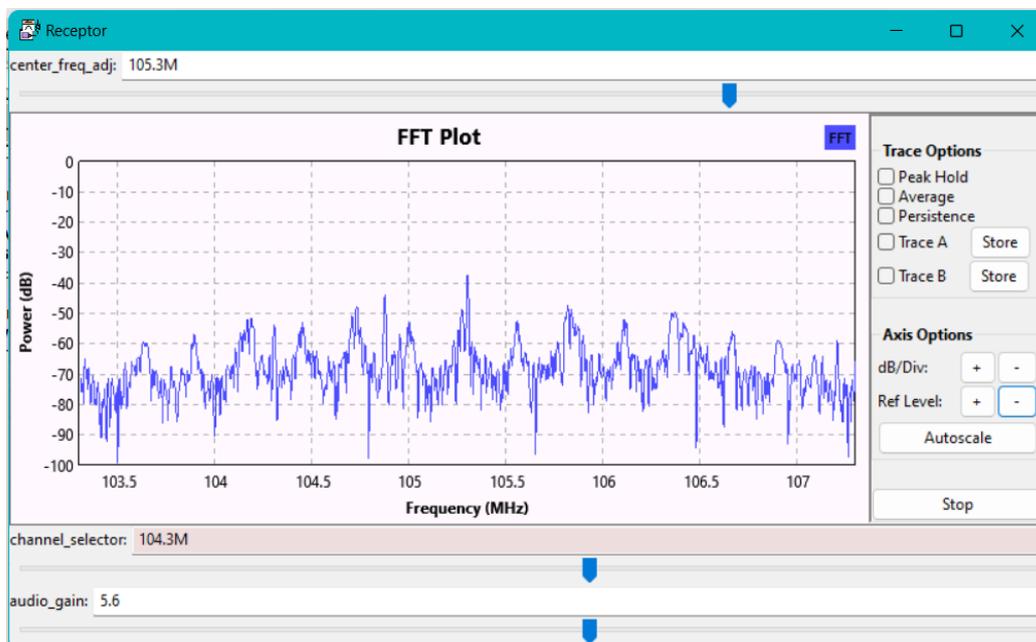
Paso 7: Por último, se ejecuta.



6

RESULTADOS**Simulación**

En el resultado, la frecuencia central con la que trabaja la recepción de la señal es de 105.3 MHz y para la demodulación del audio el canal selector se ajusta a 104.3 MHz, variando u estableciendo la ganancia del audio en 5.6 dB. Además, se puede observar en la gráfica que se utiliza la Transformada Rápida de Fourier (FFT) para obtener las componentes espectrales de la señal de audio recibida, donde se puede observar que la potencia de cada frecuencia con la que está conformada el audio oscila entre -35 a -100 dB.



7

CONCLUSIONES.

En conclusión, se realizó la guía didáctica dando una explicación exhaustiva del funcionamiento de los bloques lógicos y las simulaciones en el desarrollo de la propuesta, permitido que los estudiantes de las asignatura de procesamiento digital de señales de la

carrera de telecomunicaciones, se les facilite la comprensión y aplicación de estas modulaciones con el equipo SDR HackRF One.

8	BIBLIOGRAFIA.
	<p>[1] J. G. C. Santiana, «Universidad Católica De Santiago De Guayaquil,» 30 Octubre 2014. [En línea]. Available: http://repositorio.ucsg.edu.ec/bitstream/3317/2738/1/T-UCSG-POS-MTEL-28.pdf. [Último acceso: 23 Diciembre 2023].</p> <p>[2] «Caítulo 2: transmisión y recepción,» La Catarina UDLAP, [En línea]. Available: http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/escobar_a_g/capitulo2.pdf. [Último acceso: 23 Diciembre 2023].</p> <p>[3] «MODULACIONES ESPECIALES,» [En línea]. Available: https://catedras.facet.unt.edu.ar/ft/wp-content/uploads/sites/123/2017/03/10-MODULACIONES-ESPECIALES.pdf. [Último acceso: 23 Diciembre 2023].</p> <p>[4] Greatscottgadgets, «Greatscottgadgets.com,» 2009 - 2021. [En línea]. Available: https://greatscottgadgets.com/hackrf/one/. [Último acceso: 23 Diciembre 2023].</p> <p>[5] Lab401, «Lab401,» 1, 2023. [En línea]. Available: https://lab401.com/es/products/hackrf-one. [Último acceso: 23 Diciembre 2023].</p> <p>[6] «Gnu Radio,» 2022. [En línea]. Available: https://www.gnuradio.org/. [Último acceso: 23 Diciembre 2023].</p>